Learning about User Interface Design through the use of User Interface Pattern Languages

A thesis dissertation presented in partial
fulfilment of the requirements for the degree of
Doctor of Philosophy in Computer Science
at Massey University,
New Zealand.

Elisabeth-Ann Gynn Todd

2010

Abstract

The focus of this research is to investigate the potential of user interface (UI) pattern languages in assisting students of Human Computer Interaction (HCI) to learn the principles of UI design.

A graphical representation named a UI-pattern model was developed. It arose from the evaluation of four existing pattern languages. The UI-pattern model is an enhanced form of UI pattern list that represents a specific UI. It was recognised that the UI-pattern model has the potential to help students learn about pattern language structure. It was also realised that UI-pattern modelling can be used to incrementally improve pattern languages through the generative process proposed by Alexander (1979). A UI pattern language Maturity Model (UMM) has been developed. This model can be used by educators when selecting and/or modifying existing UI pattern languages so that they are more appropriate for student use.

A method for developing detailed UI designs that utilises a UI pattern language has been developed with the aim of providing students with an ‘authentic’ real-world UI design experience, as envisaged by constructivist educational theory (Jonassen 1999). This UI design method (TUIPL) guides the students’ development of user interface conceptual models. To establish the authenticity of TUIPL three case studies were undertaken out with developers who had differing levels of UI design experience.

A series of studies investigated how HCI students used TUIPL to guide the development of UI-pattern models and canonical abstract prototypes. The studies also ascertained the students’ views on using three different forms of UI pattern (illustrated, narrative and diagrammed). Data was collected by observation, questionnaires and completed exercises. The results indicate that the students developed an understanding of pattern language structure, were positive about their experience building UI-pattern models and canonical abstract prototypes, and that patterns aided communication. The learning outcomes were encouraging and students responded positively to using a UI pattern language.

Acknowledgements

I wish to thank A/Professor Elizabeth Kemp, my chief supervisor, for her patience and support throughout this research. Her continued encouragement and guidance helped me maintain interest in the research.

I would also like to thank my second supervisor A/Professor Chris Phillips for continuing his input even after retiring. His assistance in reading and correcting the thesis as well as providing ideas for improving layout and structure were greatly appreciated.

On a personal level, without my husband's conviction that I had the ability to complete a PhD, I would not have made it. Arthur's love and support has been pivotal for me continuing the research through the years. Thank-you Arthur, you are a treasure.

Finally I wish to dedicate this thesis to my parents Joan and Jim. For my father who wished he had attended university but died far too young and did not attain that goal. And to my mother who kept the vision alive for me, encouraging both myself and my sister to attain the best we could achieve.

Contents

Volume 1

Volume 2 – Appendices

Figures and Tables

Figures

Tables

Chapter 1: Introduction

When engineers design a solution to a well-known problem they normally refer to standards, guidelines, or templates presenting accepted models for solving that class of problem. Civil engineers will refer to predefined bridging solutions for similar foundation conditions, flood flows and anticipated traffic loadings when designing a crossing for a watercourse. The tool that fills this role for software engineers is the design pattern. These describe known good solutions for specific types of software design problem. Patterns present widely acknowledged good practice to guide many development tasks. Salingaros (2000) identifies pattern languages as useful because they are:

> *"a way of understanding, and possibly controlling, a complex system … [they are] necessary design tools with which to build something that is functionally and structurally coherent" (ibid, p154)*

Controlling complex systems and building a functional and structurally coherent system are basic requirements of software engineering (SE), which may explain why software engineers were early adopters of the concept of patterns for use in program development (Gamma *et al.* 1995, Coplien & Schmidt 1995). The SE design solutions captured in SE patterns were quickly accepted and the names of these patterns became part of the everyday vocabulary used by software engineering professionals and also began appearing in text books (e.g. Alur *et al.* 2001). To date no one user interface (UI) pattern collection has had the same impact upon the UI design community although a number of well known collections are available on-line (Leacock *et al.* 2005, Tidwell 1999, van Welie n.d.)

UI patterns were rendered visible to the UI developer community when Jennifer Tidwell's (1999) paper "Common Ground" became available on the web, although there are earlier references to UI patterns. For example, Hooper (1986) compares UI design to architectural design and suggests that UI patterns could be a useful tool in disseminating UI design knowledge. Beck and Cunningham (1987) published what is possibly the first UI pattern language consisting of five linked patterns for designing a windows-based interface. Barfield *et al.* (1994) report on the development of a UI design degree program that includes a number of interrelated courses that have the study, creation and use of UI patterns at their core.

UI professionals recognised that UI patterns have the potential to guide UI development, for example as discussed at UI pattern workshops at CHI conferences (1997, 1999). Recently publications consisting mostly of UI-related pattern languages have appeared both as books (Borchers 2001a, van Duyne et al. 2003, Graham 2003, Schümmer and Lukosch 2007, Tidwell 2006) and as web sites (Quince: quince.infragistics.com/ux-design-patterns.aspx, Yahoo! Patterns: developer.yahoo.com/ypatterns/index.php, Patterns in Interaction Design: www.welie.com/index.php).

User interface design incorporates knowledge from a range of different disciplines. It should be grounded in user-centered design (UCD) and is highly iterative (Preece *et al.* 2002). Many UI designers have investigated the integration of UI patterns into the UI development process (Borchers 2001a, Constantine *et al.* 2003, Mahemoff 1999, Molina *et al.* 2003, Tidwell 2006, van Duyne *et al.* 2003, van Welie 2001, Wilkins 2003). Seffah and Gaffar (2007) consider UI patterns to be a central and integral component for model-driven UI design saying:

> *"UI design patterns have the potential to provide a solution to the reuse problem while acting as driving artifacts in the development and transformation of models." (ibid, p1409)*

Developing user interfaces is a creative process that should be guided by UI design principles. Shneiderman (1998) identified three principles of UI Design. Principle 1 is to recognise the diversity of humans who will be involved in using the UI, the diversity of UI design team members, the diversity of tasks to be carried out and the many different interaction styles. Principle 2 defines eight golden rules of interface design:

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialogues to yield closure.
5. Offer error prevention and simple error handling.
6. Permit easy reversal of actions.
7. Support internal locus of control.
8. Reduce short-term memory load. (ibid, p74-75)

Principle 3 is to prevent errors. Many UI patterns encapsulate aspects of these principles in their '*context*, *problem* and *solution*' framework, for example Tidwell's pattern 'Forgiving Format' (2006) relates to Shneiderman's Principle 3 because it helps in reducing errors. It also relates to his Principle 1 because users can use their preferred format, for example in entering a date. The principles can also provide the basis of the discussion section that provides support for a UI pattern's solution.

UI patterns should provide a common language, a *lingua franca* (Erickson 2000), usable by all those with an interest in a UI design project. As yet this goal has not been attained. UI patterns have been reported to aid communication, for example between software engineers and UI designers (Borchers 2000, Acosta & Zambrano 2004), or between end users and developers (Dearden *et al.* 2002a, Guy 2003, Schümmer *et al.* 2006, Schümmer & Lukosch 2007). Patterns have also been shown to provide an acceptable way for non-experts to access knowledge about a specific domain. Examples include introducing adults to concepts of computer science as part of community-based activities (Carroll 2002), and communicating knowledge of architecture to builders and members of the general public (Saunders 2002). Using UI patterns to guide UI design empowers both users and developers, allowing them to identify and evaluate solutions in areas where they lack expertise (Lukosch & Schümmer 2006).

UI design is a complex problem-solving activity. As Barfield *et al.* (1994) observe:

> *"There are checklists and guidelines, rules-of-thumb and standards, specification languages and methodologies but although these are useful, they are not the solution to design problems" (ibid, p63).*

Although user interface patterns have the potential to enhance and enrich the field of human computer interface (HCI) design, as pointed out by Dearden and Finlay (2006) in their comprehensive review "*Pattern Languages in HCI*" there is a

> *"… lack of substantive evidence of their benefits for actual design practice" (ibid, p86)*

They concluded their review by identifying four important areas for future research:

- Exploring appropriate ways to use pattern languages in design and education,
- Finding ways to organise pattern languages in HCI,

- Exploring and improving the processes by which patterns are identified, recorded and reviewed,
- Examining the way that values are explicated and promulgated in pattern languages and pattern-led design.

The research reported in this thesis focuses on the first item of Dearden and Finlay's research agenda given their statement that:

> *"Significant effort is now required to examine the use of these languages … in education to demonstrate what benefits might be gained from a patterns approach" (ibid, p86).*

The immediate intention was to identify the attributes required of a pattern language that is oriented towards teaching the principles of user interface design, and using the patterns in the UI pattern language to help students develop conceptual models of interfaces. Other aspects of Dearden and Finlay's agenda also affect this intention.

1.1 Patterns and Pattern Languages

Patterns and pattern language concepts are derived from architecture. They were first proposed by Alexander as a result of his PhD studies completed in 1968. Patterns did not come to the notice of a wider readership until the publication of the first pattern language (Alexander *et al.* 1977) henceforth referred to as the Alexandrian[1] pattern language. Individual patterns can be thought of as three-part rules, which not only capture a solution to a problem taking into account identified constraints, but also explain the relationship between the three parts and how the pattern balances any conflicts between the constraints. Alexander's (1979) description clearly identifies this rule structure for a pattern:

> *"Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution…each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves" (ibid, p247).*

A pattern of this kind can be given a name and should have an associated diagram illustrating the essential elements of the solution. As Alexander says:

[1] The term Alexandrian is used to indicate an association with the writings of Alexander and his collaborators.

"If you can't draw it, it isn't a pattern" (ibid, p267).

Although the Alexandrian patterns were written for both ordinary people and professional architects or town planners, they have largely been spurned by the experts (Dovey 1990, Saunders 2002). Dovey (1990) reviewed the Alexandrian pattern language approach to environmental design presenting many of the arguments against patterns. Although rather theoretical, this paper makes several useful observations about patterns and their potential use. Dovey observes that pattern users equate the sensitising illustration with pattern content, potentially misinterpreting the pattern. For example, the illustration heading the pattern '180 WINDOW PLACE' may be confusing because:

"The distinction between looking at a window place and sitting in it has been ignored" (ibid, p4)

He also highlights that there is a cultural bias in the Alexandrian patterns because they are primarily grounded in the architectural history of the western world but observes this is a difficulty with many systems. Overall, Dovey was positive that the patterns approach actively encouraged users to participate in the design process with its recognition of many facets of design from social to physical. He concluded that architects should pay more attention to the Alexandrian pattern language, saying:

"The pattern language is one of the few current architectural theories that offers a potential theoretical ground for a world faced with severe problems of physical and social ecology." (ibid, p8)

Over a decade later, Saunders (2002) reviewed the book "A Pattern Language" (Alexander *et al.* 1977) with reference to its status in the architectural domain and the extent of its uptake within the wider community. Like Dovey (1990) he reports that there is a love-hate relationship with Alexander's work from those peers who have read his work. He calls attention to the fact that most architectural students have not been introduced to Alexander's work, yet he says "*it could be the most widely read architectural treatise of all time"* (ibid, p1). He supports this statement by commenting that he had always found the book at the top of the 'Bestselling' list in Amazon's Architecture category and recent checks (December 2009) confirm that it had retained its place in the top ten books on this list. The Alexandrian pattern language has been embraced by the user community with many an ordinary person using the Alexandrian pattern language as their "design bible" when building a new home (Finlayson 2005).

Dearden *et al.* (2002b) hint of a tension within the software domain that is almost the opposite to that found within architecture when they say that:

> *"However, the approach to pattern languages adopted within HCI has followed closely that of software engineering, with the emphasis on sharing knowledge between professionals rather than on processes to support user participation in design" (ibid, p104)*

They have worked with users in participatory design projects and have evaluated the use of different pattern forms with users designing user interfaces. They want users to experience using patterns in much the same way that Barbie and James Wilson did when building their home in Marlborough, NZ (Finlayson 2005).

1.2 Introducing Students to UI Patterns

About the time that UI designers were discovering patterns (Hooper 1986), Davis (1986) identified Alexander's patterns as a suitable tool for helping students work collaboratively on an architectural design problem. He considered each pattern to be a distinctly packaged combination of theory and practice that students could use to aid their decision-making discussions when working on a group design project. It therefore seems reasonable to assume that UI patterns would be helpful in aiding students learn about user interface design. Barfield *et al.* (1994) proposed this use in their comprehensive UI design curriculum.

Since that publication there is further evidence that introducing students to UI design patterns[2] can improve the quality of their designs (Kotzé *et al.* 2006, Laakso *et al.* 2000, Koukouletsos 2009). Borchers (2002) introduced students to UI patterns and reported that they thought that UI patterns helped them to understand how to complete their prototypes and better comprehend the UI design concepts they were applying. Griffiths and Pemberton (2004) investigated different ways in which patterns might be used in an educational setting and identified:

> *"three approaches to incorporating design patterns into teaching: teaching about pattern language, teaching through pattern language and discovering patterns" (ibid, abstract)*

There can be problems when introducing UI patterns as an aid to UI design, for example Kotzé *et al.* (2006) reported that new users of pattern sets have difficulty in

[2] Example UI patterns can be found in Appendix A8.

understanding the relationships linking patterns. Chung *et al.* (2004), whose research included graduate students, reported that illustrations were especially important when searching for a pattern to help solve a specific problem. Dearden *et al.* (2002a) observed that some users depended "*exclusively on the illustrative examples*" [ibid, p665] when selecting patterns.

Barfield *et al.* (1994) described how they introduced students to authoring their own UI patterns. Koukouletsos *et al.* (2009) developed a short course introducing students to using UI patterns. But, none of the above studies report on methods for helping students to understand and use pattern language structure. All of the investigations involving students appear to have used either lo-fidelity or high-fidelity prototyping (Rettig 1994). None used UI patterns to introduce students to conceptual UI modelling. Barfield *et al.* (1994) provide the most complete description of the pattern-guided UI design method they used with students. This aimed to

> *"... complement more systematic and conventional design methods, taught elsewhere in the curriculum, with a method that is intuitive as well as analytical in character, inspired by Alexander's approach and phenomenology." (ibid, p75)*

Their method uses scenarios, images, prototypes, time-based sequences and patterns to capture the results of analysis.

1.3 Research Goal and Questions

This research is essentially an exploratory study guided by an over-arching goal which is to investigate whether:

> *A user interface pattern language can guide student developers in the creation of conceptual user interface models.*

This goal raises a number of issues that need to be addressed. Firstly, there is a need to identify the characteristics of a UI pattern language. To teach about UI pattern languages it is necessary for the educator to understand them. Secondly, the research goal is based on the assumption that UI patterns are a suitable tool to use with students; to learn *about* UI patterns and UI pattern languages, and to learn UI concepts *through* using patterns. Thirdly, there is the implication that a method exists which enables UI patterns to be used to guide students in developing conceptual models of user interfaces.

Therefore three primary questions the research needs to address are:

Question One: What are the characteristics of a UI-pattern language?

There has been some progress in providing mechanisms for evaluating individual patterns (e.g. PLoP workshops and CHI workshops). Although there has been little guidance in identifying the defining characteristics of a pattern language, a number of authors have identified and discussed some attributes that describe pattern languages (Alexander 1979, Alexander *et al.* 1977, Fincher 1999, 2003a, 2003b, Graham 2003, Mahemoff & Johnston 2001, Mullet 2002, Noble 1998, Salingaros 2000, Schümmer 2003a, van Welie & van der Veer 2003). Identifying the relevant characteristics of a pattern language should help educators identify sets of patterns that both enhance their instructional approach and meet their teaching aims and objectives.

Question Two: Are there any specific requirements of a UI pattern language when used for teaching student UI designers?

To answer this question, students' preferences, learning behaviour and performance using alternative forms of UI pattern within a pattern language structure need to be evaluated. A conceptual UI design model diagram is required that will help students focus on the essential elements of a UI pattern's solution, not on a physical implementation as illustrated by snapshots of screens. Although being able to sketch a solution is fundamental to patterns (Alexander 1979) only a few collections of UI patterns include a form of abstract sketchable diagram to help a user create a conceptual UI design (e.g. Constantine 2003b, de Paula & Barbosa 2003; van Duyne *et al.* 2003).

Underlying this research question is the presumption that students can learn the information contained within the patterns. This needs to be investigated. As well, investigations should identify attributes of UI patterns and pattern languages which an educator could modify when introducing students to patterns. This should help students more easily make the transition to using existing collections of UI patterns, thereby effectively scaffolding student learning (Wood *et al.* 1976).

Question Three: How can student UI designers be successfully guided in the use of a UI pattern language for developing conceptual UI design models?

Educational research shows that constructing something new to represent what is being learned, enhances learning (Jonassen 1999). Expecting students to build a domain-specific pattern language, or even selecting patterns from an existing UI pattern collection to represent a new user interface before developing their knowledge of UI pattern languages appears to be too big a step (Kotzé *et al.* 2006). Therefore, an intermediate technique is required that uses a similar approach to that for developing a new user interface but allows the student to concentrate on using the pattern language within the design method.

A pattern-guided detailed UI design method is required that is grounded in constructionist educational theory (Jonassen 1999), where students are encouraged to build alternative representations of what they are studying. A philosophy student learning about Aristotle's discourse "*Politics: A Treatise on Government*" could be encouraged to construct a rich picture representing that treatise. A computer science student studying an object oriented (OO) program could be encouraged to construct the associated UML class diagram (Kemp *et al.* 2007).

1.4 Research Approach

The research starts by identifying a set of attributes recognised in the literature as characterising a pattern language. These are evaluated and refined, leading to the discovery of the UI-pattern model. This is recognised as having potential for helping students understand UI pattern language structure. The investigation into the structure of UI pattern languages leads to the development of a UI pattern maturity model.

The pattern-guided UI design method is presented as a way to provide students with an authentic UI design experience using a problem-based learning (PBL) strategy. To test these proposals the exploratory stage of a design study has been developed, composed of a sequence of three studies. They are exploratory in nature, forming the early stages of a 'Design Study' (Shavelson *et al.* 2003).

Study One (the first iteration) is primarily a quasi-experiment testing different forms of UI pattern. It also investigates the impact of the instructional artefacts and methods on student learning. The aim for this study is to 'Explore the use of a UI pattern language in the creation of UI pattern models by student UI developers'.

The aim of Study Two is to 'Explore the use of a UI-pattern language in the creation of conceptual UI models by student UI developers'. Study Two is divided into two parts. Part One applies selected lessons learned from Study One to investigate whether these changes improve the instructional artefacts. Part Two trials the next stage of the design method and is a pre-experiment to establish the viability of the conceptual UI modelling method and the instructional artefacts. A pattern-guided UI design method is proposed. It is used to introduce the students to UI-pattern models in the first two studies.

The aim of Study Three is to explore whether the proposed UI-pattern guided design method will provide students with an authentic UI design experience. A study using three UI design professionals investigates whether they can successfully use the proposed method to create models for a new UI from the specifications.

1.5 Thesis Structure

The structure of this thesis loosely follows tradition. Chapter Two is the main literature review. Chapters Three and Four investigate issues related to the structure of UI pattern languages. Chapter Five presents relevant educational issues and introduces the UI design method.

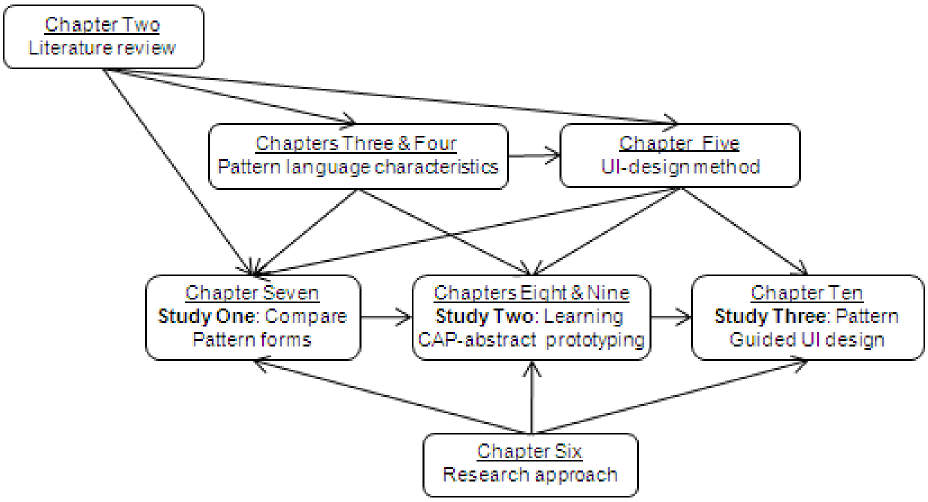


Figure 1.1 – Overview of relationships between chapters' content

Figure 1.1 shows the interrelationships and dependencies between these chapters and those of the exploratory phase of the design study. Chapter Six outlines the research approach for the design study which is reported in Chapters Seven, Eight, Nine and Ten. Finally there is the discussion bringing together the results from the different iterations and then the concluding chapter.

1.6 Thesis Outline

Each chapter is introduced in turn:

Chapter One introduces the research. It includes a brief overview of patterns and pattern languages and an indication of the potential for using UI patterns in HCI education. The research questions are introduced and an overview provided of the approach used to address these questions.

Chapter Two is the traditional literature review, providing an overview of teaching UI design using UI patterns and UI pattern languages. The chapter begins with an introduction to the use of patterns in UI design and goes on to summarise the literature about using UI patterns and pattern languages in HCI education. The potential benefits of using UI patterns are identified. The second half of the chapter considers issues that need to be addressed when using UI pattern languages as a guide to Teaching UI design. Finally, three factors identified in the literature as being significant for student learning and requiring further research, are listed.

Chapter Three investigates the characteristics of the internal structure of UI pattern languages. Before a pattern language can be successfully introduced to students the educator needs to understand it. The chapter begins with a review of the literature concerned with pattern language validation. The outcome is a proposal for a set of seven tests which assess the internal validity of a UI pattern language.

Chapter Four builds on the results of Chapter Three and re-examines the set of tests formulated to assess the internal validity of a UI pattern language. These tests are used to develop the first version of a pattern language maturity model that can be used by educators to analyse and compare existing collections of patterns. A second outcome of this investigation is the UI-pattern modelling technique. This is recognised as a potential method to scaffold students' learning about the structure of UI pattern languages.

Chapter Five introduces a model-driven pattern-language-guided UI design framework specifically oriented to introducing the students to UI patterns and pattern languages, and using them to build UI conceptual models. The methods from the framework were designed to provide students with an authentic UI design experience that can be used in the context of problem-based learning (PBL). The chapter begins with a brief review of the PBL literature that directly relates to the research and introduces the constructivist approach to learning and the technique of scaffolding student learning. The initial versions of the UI pattern language tailored for teaching students is also introduced.

Chapter Six introduces the three studies that make up the core of the research. The main research approach used in each study is described and the associated data collection and analysis methods are identified. The chapter brings together the artefacts used in the studies and identifies any variations. The procedures to be followed during each study are also introduced.

Chapter Seven presents the first study, identifying its aims and the details of the within-subjects design. The primary aim is to investigate the impact of illustrative examples in a UI pattern on student learning and acceptance. More than one data collection method (observations, model evaluation and questionnaires) is used in this study and the results of each are described in turn. A discussion triangulates the results from the four sets of data. Based on these findings, changes to the artefacts and the introduction, aimed at improving student learning are proposed.

Chapter Eight introduces Study Two which is in two parts. The first part replicates the first exercise used in the first study, but with changes in pattern format, demonstration example and pattern language structure. This implements the key recommendations for change from the first study. The results are presented in a similar manner to that used in Study One. They include a comparison of the results from the first exercise in both studies to gain insight into measured and perceived differences.

Chapter Nine presents the second part of Study Two. It is a pre-experiment to trial the viability of the second phase of the proposed pattern-guided UI design method. This exploratory investigation aims to discover whether students can learn to use canonical abstract prototyping components with navigation components to create UI conceptual models. This model has been proposed as suitable for introducing students to conceptual modelling. The discussion brings selected findings from Studies One and Two together to determine the degree to which the second research question has been answered.

Chapter Ten presents the case studies forming the third study. First the protocol used for each of the case studies is presented including an explanation of modifications based on the pilot studies. After describing the characteristics of the participants the cross-case comparison is presented. The findings of the participants' experiences with the modelling and prototyping techniques used in the second study are presented. Their views on the influence the tools and techniques may have on communication within a UI design team follow. Next, their opinions of UI-patterns as aids to UI design are presented. The findings are then discussed to determine whether the objectives of this study are achieved. A final discussion determines the degree to which the third research question has been answered. The results from consolidating and triangulating the findings from all the preceding studies are presented.

Chapter Eleven begins with a discussion of whether the three primary research questions have been answered to determine whether the research goal has been addressed. Then it presents the conclusions derived from the research project. The main research outcomes and findings are presented followed by recommendations for educating HCI students about and through UI pattern languages. The chapter concludes by identifying areas for future research.

Chapter 2: Literature Review - Pattern Languages and Teaching UI Design

This chapter reviews pattern languages in the context of their use in education with a focus on identifying the attributes required of a teaching oriented UI pattern language.

After an introduction to relevant literature covering UI patterns and their use by UI developers in designing UI, the literature dealing specifically with the use of patterns as they relate to education follows. Potential benefits of using patterns in an educational setting are identified. This leads to the identification of some attributes required of a UI pattern language that is oriented to teaching the principles and practice of user interface design. Issues of concern include the form and content of UI patterns, how a pattern language should be organised with particular reference to the type of references linking patterns and the forces they resolve. Finally future directions for research into appropriate ways for using pattern languages in UI design teaching are proposed.

2.1 Patterns and Pattern Languages in UI Design

There is a growing interest in the different kinds of patterns associated with UI design. This research is concerned mostly with interaction design patterns (Dearden & Finlay 2006) which illustrate the issues underpinning user interface design decisions. The definition of UI design patterns by van Duyne *et al.* (2003), states that:

> *"Patterns communicate insights into design problems, capturing the essence of the problems and their solutions in a compact form. They describe the problem in depth, the rationale for the solution, how to apply the solution, and some of the trade-offs in applying the solution" (ibid, p19).*

An alternative definition by Borchers (2001a) is:

> *"… a structured textual and graphical description of a proven solution to a recurring design problem" (ibid, p8).*

These two definitions complement one another. van Duyne *et al.* focus on the content of a pattern and stress its use as an aid to communication while Borchers identifies that the presentation of a pattern's content is structured and includes illustrative elements. Both authors emphasize that patterns provide the reader with a solution to a recognised problem. Patterns then can act as a *lingua franca* that is a communication device and a means of illustration when discussing interface design (Bayle *et al.* 1997, Erickson 2000, Griffiths & Pemberton 2001, Marcus 2004, Dearden & Finlay 2006). When linked together in a pattern language, relationships between the patterns should be synergistic, so that the language provides an understanding of the UI design domain that is greater than just the application of the individual patterns.

In developing an educational framework to integrate human-centred design techniques into software engineering (SE) curricula Seffah (2003) sees UI patterns as:

> *" a vehicle for highlighting common user problems and for describing proven design solutions to fix them" (ibid, p42)*

He describes the use of patterns as both resources for student learning and as an aid to UI design. He identifies three kinds of interaction pattern:

1. Product patterns which describe what the interface or interaction should be like. Most of the available collections of UI patterns fall into this category (e.g. Tidwell 1999, 2006, van Duyne *et al.* 2003, van Welie & Klaassen 2004, Schümmer & Lukosch 2007).

2. Process patterns which describe UI design methods.

3. Pedagogical patterns which describe effective learning and teaching strategies. Seffah (2003) is not alone in identifying patterns used for disseminating pedagogical theory and practice to future educators (Carroll 2002, Frizell & Hübscher 2002, Garzotto *et al.* 2004).

The UI design patterns that are the focus of this review can be classified as product patterns. These patterns fit the vision of Erickson (2000) of a *lingua franca* so that all interested can participate on an equal footing in the UI design process, rather than those patterns referred to as creating a *technical lexicon* (Dearden & Finlay 2006). The latter contain technical content restricting their use to participants with professional knowledge.

Many authors have developed one or more sets of related patterns, variously referred to as languages, catalogues or collections. A number of authors have based their collections of patterns on industrial experience, developing a large number of diverse user interfaces (Green *et al.* 2003, Laakso & Laakso 2004, Tidwell 2006). Anecdotal evidence from comments on web sites about the usefulness of van Duyne *et al.*'s (2003) book supports a belief in the usefulness of patterns to provide guidance in the creation of useable interfaces. Whilst based on Alexandrian ideas, most authors use their own UI pattern form.

2.1.1 UI Pattern Forms

Henninger and Corrêa (2007) surveyed over 2000 patterns related to software development accessible via the web. They identified fourteen collections of UI patterns totalling four hundred and twenty-five patterns but they also identified six major problems hampering the potential this resource may provide. They noted that it is difficult to have a *lingua franca* for discussing patterns and the relationships between them when there are so many different formats for presenting UI patterns.

Source	**Domain**	**Sections within the pattern description**	**Element Types**
Borchers (2001)	HCI	name, rank, illustration, context, problem, rationale & forces, solution, diagram, reference	text image sketch
Graham (2007)	HCI Usability	name, number, alternatives, sensitising image, context, problem, forces, solution, resultant context, contributors or sources	text image
Laakso (2003)	UI Design	name, a body of narrative description for the pattern, examples and references	text image
Leacock, Malone, & Wheeler (2005) Yahoo	UI for Web applications	title, author, contributors, problem summary, sensitising example, use when, solution, rationale, accessibility, supporting research, related patterns, as used on Yahoo!, UI code examples	text image animation code
Little, A of Infragistics (2009) Quince	User experience patterns	name, sensitising image, problem, solution context, rationale, implementation, examples, sources, tags	Text, images
Schümmer & Lukosch (2007)	Computer-mediated interaction	name, confidence, alternative names, thumbnail, dictionary definition clarifying name, intent, context, problem, scenario, symptoms, solution, dynamics, rationale, check, danger spots, known uses, related patterns	text image sketch
Tidwell (1999)	UI Design	name, examples, context, problem, forces, solution, diagram, resulting context, notes	text sketch image
Tidwell (2006)	UI Design	name, picture, what, use when, why, how and examples	text image
van Duyne *et al.* (2003)	UI for Web applications	name, picture, background, problem, forces, solution, and other patterns to consider	text image sketch
van Welie (2003)	User Interaction Design	name, author, problem, principle, context, forces, solution, rationale, examples, counter examples, known uses, related patterns, implementation	text image code PLML
van Welie (2007)	User Interaction Design	name, problem, solution, picture, use when, how, why, more examples, implementation, literature	text image code PLML

Table 2.1 - A selection of UI pattern forms identifying the element types

Although UI design patterns have many forms the essential parts of these patterns are the context, the problem and the solution. Table 2.1 shows the variety of forms employed in thirteen different UI pattern collections. Only Borchers' HCI patterns conform to Alexandrine practice, otherwise explicit labels are used to identify each part of a pattern. These examples also show that there are various ways of referring to the different parts of a pattern. Some use Alexander's terminology 'context', 'problem', and 'solution' with others using terms like: 'use when', 'why' and 'how'. They all contain text based descriptions but there is no commonality in the other types of element that are used. Most of these pattern forms have some visual element illustrating different aspects of a pattern. These can be one or more of images, sketches or animations. Some even include code examples. Another problem is the digital form in which these patterns are available varies from: text, pdf, html and forms based on XML.

2.1.2 Using UI Patterns in Design

A number of studies have investigated the use of patterns as part of the UI design process. The participants in these studies were portrayed as potential participants in UI design teams although in some, students were included.

Dearden *et al.* (2002a, 2002b) report on investigations into the use of patterns in participatory design where their participants develop UI designs for a website. They recorded substantial differences in the time participants spent studying individual patterns. Their observations also revealed that different parts of patterns took the attention of different individuals. They noted that most of the participants did not read the associated explanations. This behaviour was summed up by one participant stating "*the patterns were just too wordy*" (Dearden *et al.* 2002b, p108). The authors remark that many existing UI patterns were written for designers using language with which an experienced UI designer would be familiar, but not necessarily a novice designer.

Furthermore, they observed that some users depended "*exclusively on the illustrative examples*" (Dearden *et al.* 2002a, p665) when selecting patterns. One user actually went as far as "*equating the pattern with the example picture*" (Dearden *et al.* 2002b, p108). They suggest that the prominence of the illustration used in the patterns may have been a factor in some users' reliance on them. Some of the participants "*suggested alternative layouts*" (ibid, p108) which may overcome this problem. Dearden *et al.* (2002a) found that the physical form of the pattern could influence how users reacted to using patterns in the design process. Similarly, Sharp *et al.* (2003) in a study involving pedagogical patterns found that it would be desirable for patterns to be formatted in ways that better suit a specific audience completing a specific task. Even though problems were observed during their experiment (Dearden *et al.* 2002b), in the final evaluation participants were positive about using the patterns, reporting that "*they found the patterns helpful*" (ibid, p111).

Finlay *et al.* (2002) summarise a set of experiments including the one reported in Dearden *et al.* (2002a, b). The follow-up study extended their investigations into participatory design focussing on helping the participants use the structure of the pattern language. This experiment had similar findings to the earlier one with regard to how the participants used the individual patterns. They also found that by having the facilitator emphasise the links between patterns when introducing the next pattern, that by the end of the design session, participants selected patterns based on the links without help. One user stated "*once I got used to using the links to go to different patterns, it's quite useful*" (ibid, p170). This experiment had two stages and users were observed paying more attention to pattern content and using links for cross referencing in the second session leading to the suggestion:

> *"... that users' needs may change as they become familiar with the patterns, and the form should support this" (ibid, p171).*

An investigation into the suitability of UI patterns compared to UI guidelines as part of an heuristic evaluation of a website identified a number of issues the two authors had with using patterns (Wesson & Cowley 2003). Some of these were not dissimilar to those identified by Dearden *et al.* (2002a, 2002b), for example the patterns needed to be studied before they could be used and there was confusion over the links between patterns structuring the pattern set. To help understand the links the participants created a pattern map. They also considered that some pattern names hindered their use.

Similar issues to those reported above were identified in an investigation into the use of pre-patterns to inform designers unfamiliar with ubiquitous computing in creating a design for location-enhanced applications (Chung *et al.* 2004). Pre-patterns present information using a pattern format but describe problems with solutions in a domain which is so new that best practice has not yet been established. They carried out a series of experiments which included both novice and experienced designers. The patterns initiated communication among the participants but these participants also had difficulty with the amount of reading required and difficulty with some of the language used. Also they had a tendency to rely on the illustrations of examples. The importance of illustrated examples was also highlighted. Chung *et al.* (2004) reported that participants used the patterns by:

> *"... skimming through the names and pictures to come up with ideas" and "used the patterns to communicate ideas by pointing at a particular picture" (ibid, p237).*

Results showed that experience in UI design was more important than pattern use when measuring the quality of resultant prototypes. The experienced participants still believed patterns to be a helpful form for presenting UI information. Importantly for the research reported in this thesis, inexperienced designers found patterns the most useful.

Segerstahl and Jokela (2006) carried out a case study that also reveals the problems developers have encountered when using pattern collections. There were issues with the examples, pattern naming and the time taken to learn to use the patterns. They report that none of the pattern collections covered all the problem cases and that a pattern can appear in several collections with a different name, causing confusion. They also noted the lack of a standard means of organising, documenting and grouping patterns. Like others (Deng *et al.* 2006, Henninger & Corrêa, 2007) they recommend that all useful patterns from the various collections be gathered into a common UI pattern language.

Saponas *et al.* (2007) carried out a between-subjects study by compared two groups of participants working on a design problem in an unfamiliar domain. They used pre-patterns, like Chung *et al.* (2004). The focus was the design of an interface for a digital home. The participants were practicing professional designers. One group worked with the patterns while the other had no additional aids. The resultant designs from the two groups were evaluated and compared. No significant difference was observed but the designs by the patterns group had on average fewer heuristic violations. Analysis of observations identified four ways in which the designers used the patterns. Discovery occurred when participants were becoming familiar with the pattern content and identified a subset that might be useful. Idea generation had the goal of finding a solution to a high-level design problem. Issue clarification had the goal of finding a solution to a specific UI problem. Re-reference is the process where patterns are revisited to review decisions or to locate a remembered idea. Members of the patterns group used pattern names when discussing UI design ideas.

Authors	Participants	UI Patterns	Application	Findings
Dearden *et al.* (2002a, b) Experiment	6 users ranging in experience from a retired non-web-user to a trainee web designer.	23 based on Tidwell's Common Ground and Borchers HCI patterns	Patterns Individuals working with facilitator	Empowered users Patterns seen as helpful Pattern presentation critical Over-reliance on illustrative examples Importance of pattern as physical artefact
Finlay *et al.* (2002) Experiment	2 students and 4 university lecturers who all had some basic web design experience	39 (Educational and web)	Patterns focussing on pattern language structure Individuals working with facilitator	Empowered users Patterns focus of communication Pattern presentation critical Illustrations important Facilitation helped understand pattern language structure Flexible pattern form required.
Wesson & Cowley (2003) Case study	2 participants	16 e-commerce patterns – from Duyne *et al.*'s Web interface patterns	Guidelines versus patterns Individuals	Patterns require time to read and reflect on Time to understand links Developing pattern map helped understanding of links Pattern names are important
Chung *et al.* (2004) Experiment	8 Professionals 10 Graduate students	45 pre-patterns ubiquitous computing (UbiComp)	Patterns versus no-pattern groups Worked in pairs	Patterns helped novice designers Patterns helped designers with unfamiliar domain Reliance on name and picture when selecting patterns Pictures in patterns used to communicate ideas Patterns helped designers avoid some design problems
Chung *et al.* (2004) Experiment	6 Professionals 8 Graduate Students	30 simplified pre-patterns (UbiComp)	Pattern versus no-pattern groups (5 no-pattern pairs as for experiment 1 above) Worked in pairs	Patterns seen as helpful Patterns require time to absorb Experience is more important than using patterns Diagrams illustrating solutions important
Segerstahl & Jokela (2006) Case study in commercial setting	Professional UI designers	Based on patterns from Tidwell's Common Ground and van Welie's ... interaction design patterns.	Patterns UI design team	Patterns required time to absorb Illustrate examples important Pattern names can affect use UI pattern collections focus on different design problem domains
Saponas *et al.* (2006) Experiment	44 Professional Designers (UI or Industrial Designers)	48 pre-patterns in Digital Home and UbiComp based on own experience and Chung *et al.* 's UbiComp patterns	Patterns versus no pattern group Patterns browsing tool used Worked in pairs	Patterns group had lower number of violations in designs Identified 4 ways patterns were used, most used for idea generation & issue clarification Pattern names used for content communication, Designs of pattern group less detailed than others
Lin & Landay (2008) Experiment	12 Professional Designers	Design of Sites – Van Duyne at al	Within subject – Damask plus patterns, Damask without patterns Worked individually	Designers at least as fast using patterns even with browsing Desktop and Smartphone designs based on patterns rated higher than others Designers found patterns useful
Bernhaupt, Winckler & Pontico (2009) Field study in commercial setting	8 Professionals who were expected to work with patterns	SmalS-MvM in-house pattern language	Worked individually	Patterns acceptable form Background knowledge may affect understanding Patterns encourage discussion of alternative solutions Assists solution re-use Pattern language should evolve

Table 2.2 - Summary of UI design oriented studies involving UI patterns

Lin and Landay (2008) carried out a within-subjects study that investigated the use of patterns by forty-four UI professional designers in early-stage design of cross-device interfaces. This study was focussed on Damask, a tool for early stage UI design which has UI patterns embedded as part of the toolbox available to designers. The participants rated the patterns as useful. The design of UIs for desktops and Smartphones created by the patterns group were rated higher than those of the control group. The patterns describing the voice UI were pre-patterns. Like other studies involving pre-patterns (Chung *et al.* 2004, Saponas *et al.* 2007), no significant difference between the designs by the control and pattern groups was found.

Bernhaupt *et al.* (2007) undertook a field study to investigate the usage of a UI pattern language that had been specifically developed for an organisation, with the aim of improving standardisation of UIs for new e-Government applications. The organisation encouraged employees to use this pattern language to assist all members of the design team understand UI solutions. Six of the eight participants had experience using the pattern language. They were asked to carry out three tasks centered on the design of prototypes and were subsequently interviewed about how the pattern language could be improved. The authors concluded that the pattern language was seen as useful, readable and adaptable. The use of patterns helped to spread knowledge of usability within an organisation but the participants believed a pattern language must keep on evolving to have credibility. In practice, project constraints (such as time) reportedly limited use of the patterns on projects.

Table 2.2 summarises the findings from the selection of studies described above. The participants, the source of the UI patterns, the structure of the design team and the primary findings from each study are all identified. The results indicate that participants from many backgrounds responded very positively to using UI patterns (Dearden *et al.* 2002a, b, Chung *et al.* 2004, Lin and Landay 2008, Bernhaupt *et al.* 2009) and that patterns can be a positive influence on communication (Finlay *et al.* 2002, Chung *et al.* 2004, Saponas *et al.* 2006, Bernhaupt *et al.* 2009). The visual elements of patterns, such as illustrated examples and diagrams, are also seen as being very important (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Chung *et al.* 2004, Segerstahl & Jokela 2006).

2.1.3 Pattern Guided UI Design Methods

Authors who have contributed regularly to the UI pattern literature have each proposed at least one method for using patterns in the development of user interfaces (Mahemoff

1999, Borchers 2001a, Granlund *et al.* 2001, van Welie 2001, Crabtree 2002, Graham 2003, Molina & Hernandez 2003, Pribeanu & Vanderdonckt 2003, van Duyne *et al.* 2003, Walldius 2003, Wilkins 2003, Tidwell 2006). Many authors (e.g. Bayle *et al.* 1998, Erickson, 2000, Dearden & Finlay 2006, Schümmer & Slagter 2004) identify one key advantage of using patterns in design is that they support 'piecemeal development' first coined by Alexander (1979). This refers to evolutionary incremental development where each increment should be useable. Most authors describe the integration of pattern use into existing SE approaches (Wentzlaff & Specker 2006). Many of the methods proposed include the development of a pattern language for describing the specific UI problem domain that was the current focus of the development team (Granlund *et al.* 2001, Crabtree *et al.* 2002).

Where an existing collection of patterns is referenced, these methods usually give little guidance on how patterns should be selected from the collection and used. Other researchers (Schümmer & Slagter 2004, Schümmer & Lukosch 2007) adapted the method described in the Oregon Experiment (Alexander *et al.* 1980). This method is reliant on a set of over seventy patterns describing computer-mediated interactions (Schümmer & Lukosch 2007) which have links to existing collections of UI patterns (e.g. Graham 2003, van Duyne *et al.* 2003, Tidwell 2006, van Welie accessed 2007).

Seffah and Gaffer (2007) proposed a model-driven UI design method that is guided by a set of patterns for different levels of UI design. This pattern-driven model-based UI development framework (PD-MBUI) has an associated comprehensive tool based on XML and XUL. Patterns guide the creation of a series of models at different levels of abstraction: the task, dialog, presentation and layout levels of UI design. The method has three phases. In phase one, three models are created, the domain model, the user model and the task model. Phase two results in the composition of an environment model, a dialog model, a presentation model and a layout model. In the final phase lo-fidelity and high-fidelity prototypes are created. UI patterns are associated with dialog, presentation and layout models. Their method helps designers with:

> *"identifying, instantiating and applying patterns during the construction and refinement of these models" (ibid, p1411)*

Alexander *et al.* (1977) described a generative process in an eight step method which guides the user in how to select and use patterns. This method is based on using a list of the pattern names to keep track of which patterns in the language that have been selected. The users build a pattern sequence which is a list of the pattern names that describe their problem. After selecting a relevant pattern, the user examines the context and reference sections and selects from these lists the next patterns they need to consider. This sequence of patterns is effectively a meta-language for the project (Erickson 2000, Seffah & Forbrig 2002, Todd 2004). Next, the user reviews their pattern sequence and they write patterns for any missing elements. The user is then encouraged to modify the patterns in the sequence to reflect the project. In this way pattern languages grow and evolve. The sequence of interlinked patterns could be described as a pattern model for the many potential UI design solutions which could be created from it for the project.

2.2 Patterns in UI Design Teaching

Patterns have repeatedly been propounded as an aid to communication and for providing an agreeable way for the non-expert to access knowledge about a specific domain. Barfield *et al.* (1994) used the Alexandrian patterns to introduce students to developing UI patterns as part of UI design. They observed that:

> *"Alexander's patterns may even be more applicable to interactive systems than they are to buildings, since interactive systems literally have behaviours inside them (manifest system behaviours as well as evoked user behaviours)" (ibid, p71)*

Very early in the history of architectural patterns Davis (1986) identified Alexander's patterns as suitable tool for helping architectural students work collaboratively on a design problem. Davis observed that:

> *"The explicitness of the patterns makes them ideal for group work: the issues are always clear, and individual decisions are relatively un-encumbered by others" (ibid, p15)*

It therefore seems reasonable that UI patterns would be helpful in aiding students learn about user interface design. The use of patterns to successfully teach new UI concepts is supported by experience in the related discipline of SE (Clancy & Linn 1999, Cybulski & Linden 2000, Weiss 2005). Teaching about SE design patterns is now an accepted part of the SE curriculum (Shinichi, *et al.* 2007, IEEE/ACM Joint Task Force on Computing Curricula). More recently Kolfschoten *et al.* (2010) reported that they had successfully used design patterns in courses for some years. They found that using patterns:

> *"help novices to faster gain understanding in problem solving and design skills"*

and

> *"to better understand the problem domain" (ibid, p659)*

Also that novices taught using patterns developed higher-quality designs.

2.2.1 Studies of UI Patterns in Education

When introducing the experiments Kotzé *et al.* (2006) carried out to explore aspects of using patterns as a potential teaching tool they state that:

> *"One of the challenges in HCI teaching is to develop effective techniques for communicating the knowledge and experience of HCI experts" (ibid, p10.1)*

Educators (Barfield *et al.* 1994, Laakso *et al.* 2000, Borchers 2002, Griffiths & Pemberton 2004, van Biljon *et al.* 2004, Kotzé *et al.* 2006, Koukouletsos 2008) report various levels of success in using UI patterns to do just that. Barfield *et al.* (1994) present a four-year curriculum for an interaction design degree taught at the Utrecht School of Arts based on Alexander's (1979) philosophy. By introducing the use of patterns as a UI design tool they hoped to sensitise students to the needs of users. This should result in students developing high quality solutions permeated by human values. Barfield *et al.* (1994) indicate that their initial experiences and the evaluations from students were positive. Students were encouraged to develop their own patterns and develop low fidelity prototypes to demonstrate the solutions they have proposed. Prototypes encourage students to communicate their ideas about the underpinning conceptual model and required interaction detail.

Laakso *et al.* (2000) developed a pattern language specifically aimed at HCI students based on the evaluation of three hundred and fifty student projects over a four year period. This pattern language addressed recurring UI design problems that their students encountered in their assignments. The striking feature of these patterns is that they rely heavily on the illustration of examples and minimised the associated discussion. Laakso has integrated these patterns into HCI design courses with reportedly successful results. Although no statistics are given the anecdotal evidence indicates that the greatest gain was for the less experienced students who produced acceptable designs in a shorter time frame than previously.

Borchers (2001a, 2002) presented the concepts of patterns to students in an introductory HCI class via a lecture and provided them with Tidwell's (1999) Common Ground pattern set to use in a prototyping exercise. A follow-up survey revealed that a significant number of students considered that the patterns had been useful in aiding their understanding of UI design concepts and that the patterns were useful in informing their own design. He reports that

> *"these results indicate that a pattern approach in HCI education is useful and convincing … even first-year undergraduates can quickly relate to this format, and find it useful" (Borchers 2002, p195).*

Barfield *et al.* (1994), Griffiths and Pemberton (2004) and Borchers (2002) all had students develop their own sets of patterns. The results of evaluations (Borchers 2002, Griffiths & Pemberton 2004) indicate that students had little difficulty in conforming to a pattern-like template. Students did have problems in finding "*the right level of granularity and abstraction*" (Borchers 2002, p3).

The HCI patterns created were:

> *"close to generic guidelines, focusing on descriptive criteria for good user interfaces, but not providing much constructive help in the design process" (ibid, p3).*

Griffiths and Pemberton (2004) also report that the UI designs created by the students using Tidwell's (1999) Common Ground pattern set "*were considerably improved*" (Griffiths & Pemberton 2004, p2). They observed that the quality of the discussion between the students improved too.

Cowley and Wesson (2005) carried out an experiment where thirty-three Honours and Masters level students were divided into two groups using a stratified sampling approach. One group used patterns while the other used guidelines. The students completed three tasks: evaluating and then redesigning an existing web site followed by designing a new E-commerce website. Information was collected by a post-test questionnaire. For all three tasks students found patterns to be both efficient and effective. With regard to the new design task, the students rated patterns satisfying to use. However, the results for patterns and guidelines were similar. Interestingly by the end of the study students considered the patterns to be their personal design language.

An empirical experiment was conducted (Wania 2008, Wania & Atwood 2009) in which three groups of students were asked to develop a paper prototype for an interface to an information retrieval application. The participants were familiar with this domain.

One group was exposed to a pattern language, another to guidelines whilst the control group had no specified design technique. No statistical difference was found between the groups with respect to either the quality of the paper prototypes (as rated by experts) or the time taken to develop them. As Wania (2008) noted, given that the domain was familiar people might already have knowledge of relevant patterns. She reported that patterns and pattern languages appear to describe quality and usability in a comprehensive manner. A further analysis of the paper prototypes indicated that patterns were used by all groups although those produced by the pattern group on average contained more patterns. There was a positive correlation between the number of patterns identified in the prototypes and their quality rating of the UI.

Anti-patterns are a well established method for describing inappropriate solutions that are frequently repeated in the SE domain. A series of experiments have been carried out to investigating the use of anti-patterns in teaching UI design (van Biljon *et al.* 2004, Kotzé *et al.* 2006, Kotzé *et al.* 2008). The difference between presenting information as positive or negative was investigated by van Biljon *et al.* (2004). Their findings indicated that presenting students with anti-patterns could have undesirable side effects. They grounded their argument in the context of a learner's cognitive development and mental models. They show that presenting students with anti-patterns before they are sufficiently familiar with patterns at best confuses the learner or at worst results in the learner mistakenly identifying the anti-pattern as being the preferred solution.

Kotzé *et al.* (2008) includes an in-depth report on a third study where they were primarily interested in exploring whether providing negative solutions would be effective in helping students create better UI designs. In these experiments guidelines were substituted for patterns. The results provide insights into the use of patterns as a teaching tool. They found that presenting students with positive forms was more effective than providing negative ones.

Kotzé *et al.* (2006) included a discussion of a small study which compared two groups of students. One group used a small collection of patterns and the other used a set of guidelines. The authors identified naming as an important influence on how successful students were at using patterns. They observed that a poor pattern name obscured the intent of a pattern. Kotzé *et al.* (2006) highlighted the problem new users of pattern sets have with understanding the relationships linking patterns and identify the importance of visual elements in patterns used by students.

Koukouletsos *et al.* (2007, 2009) followed up the study reported in Kotzé *et al.* (2006) with an experiment involving thirty-nine final year students who had no previous knowledge of UI web design or UI usability principles. One group of students were exposed to a set of guidelines that contained the required knowledge while the second group used a set of patterns that had been created specifically for student learning and contained equivalent knowledge. Some relationships were defined between the patterns but a complete pattern map could not be created. The experiment tested the hypothesis that there would be a performance difference between the two groups of students. The experiment required the development of a half semester course covering these topics. A series of standard teaching sessions introduced usability principles to the students. A test followed to determine students' understanding of the material. The students then worked individually, developing a high fidelity prototype for a small web site.

A predetermined set of metrics was used to evaluate the web sites by three independent assessors. Unlike previous studies discussed above, the prototypes created by the patterns group were statistically significantly better than those created by the guidelines group. They also found that using patterns facilitated communication of ideas but pattern names were not perceived by the students as helpful and nor were the relationships linking patterns.

Koukouletsos (2008) reported that the students tried:

> *"… to understand the principles [by] relying mostly on the examples instead of reading the text" (ibid, p226)*

confirming findings of Dearden *et al.* (2002a, b). But, it was by embedding these examples within the text of a pattern that students perceived as particularly helpful when compared to guidelines. This focus on the illustrated examples has also been noted by other researchers (Chung *et al.* 2004, Segerstahl & Jokela 2006, Kotzé *et al.* 2006). After being exposed to both patterns and guidelines, at the end of the course responses to a follow-up interview elicited a preference by all students for the UI patterns.

Authors	Subjects	Patterns Used	How applied	Findings
Barfield *et al.* (1994)	100 students enrolled in 4 yr degree	Based on UI guidelines	Student writing own patterns Worked in groups and individually y	Identifying UI design with quality and value Sensitising students to the needs of users Linking UI design with philosophy and psychology
Laakso *et al.* (2000)	350 university students	Laakso's UI Design Patterns (adapted from Tidwell's Common Ground)	Patterns	Improved communication More efficient design process Helped students learn design more quickly Improved quality of products produced
Borchers (2001a, 2002)	26 undergraduate students	Tidwell's Common Ground	Patterns Worked individually	Students consider patterns useful for remembering UI design concepts Students thought they would use patterns in future
Borchers (2002)	10 postgraduate students 8 undergraduate students	Tidwell's Common Ground Borchers Van Welie (web)	Student writing own patterns Worked in groups and individually	Structure of patterns easily understood and applied Students avoided designs that were too concrete Students had problems with level of granularity Improved retention of UI design principles
Griffiths & Pemberton (2004)	11 undergraduate students	Relevant web sites and readings	Student writing own patterns Worked individually	Students found it difficult to generalise from examples
Griffiths and Pemberton (2004)	11 undergraduate students	Tidwell's Common Ground	Patterns Worked in groups	Improved discussion Improved quality of products produce
Cowley & Wesson (2005)	33 Masters & Honours Students	Barnard (2004) e-commerce guidelines van Welie (Web) van Duyne *et al.*'s Design of Sites patterns	Patterns versus guidelines Worked in groups	Patterns seen as efficient and effect for evaluation, redesign and design Patterns satisfying to use for designs new UI Little overall difference between patterns and guidelines groups
Wania & Atwood (2009)	37 undergraduate students 15 postgraduate students	39 Information retrieval patterns aIRPLANE	Patterns, guidelines & undirected Worked individually	No significant difference in quality of designs or time taken Higher quality associated with higher pattern use
van Biljon *et al.* (2004) Kotzé *et al.* (2008)	66 honours students – database and Internet technology	Authors positive and negative guidelines (Ecommerce)	Patterns, versus anti -patterns Worked in groups of 3	Positive pattern forms more effective Pattern structure and presentation important Previous experience can impact pattern knowledge uptake
Kotzé *et al.* (2008)	54 undergraduate students	Authors positive and negative guidelines plus examples (issues relating to design web pages)	Patterns versus anti-patterns Worked individually	Pattern group performed statistically better than anti-pattern group
Kotzé *et al.* (2006)	11 undergraduate students	Van Welie (web)	Patterns versus guidelines groups Worked individually	Guidelines easier to teach than patterns Links between patterns not appreciated by students Illustrated examples captured student attention Names important
Koukouletsos (2008), Koukouletsos *et al.* (2009)	39 undergraduate students	Koyani *et al.* (2004) guidelines Authors created patterns to match guidelines (adapted from van Welie and van Duyne *et al.*)	Patterns versus guidelines Worked individually	Patterns modified especially for students Both patterns & guidelines help develop usable prototypes Pattern guided UI models statistically better Students tend to rely on illustrated examples. Compound patterns close to student's cognitive limits

Table 2.3 - Comparison of teaching oriented studies involving UI patterns

The experiences reported above and summarised in Table 2.3, indicate that encouraging students to consult a set of UI patterns may improve their understanding of UI design (Borchers 2001a, Koukouletsos 2009) and can have a positive influence on the quality of their work (Laakso *et al.* 2000, Griffiths & Pemberton 2004, Kotzé *et al.* 2006). Teaching experience with UI patterns indicates that the introduction of patterns to students should be embedded within a design course (Laakso *et al.* 2000, Borchers 2002, Griffiths & Pemberton 2004).

2.2.2 Teaching with UI Patterns

Seffah (2003) identified UI requirements gathering via patterns and guidelines as a specific skill required by students studying UI development. Kotzé *et al.* (2006) points out:

> *"patterns seem to require more careful and thoughtful teaching approaches if we want to realize their full potential" (ibid, p10.5).*

But, few of the studies (Barfield *et al.* 1994, Koukouletsos 2008) discussed processes for introducing the students to UI design pattern languages.

The most comprehensive discussion of how to introduce students to UI patterns is provided in the curriculum discussion by Barfield *et al.* (1994). They designed a set of three integrated courses that focus on a pattern-driven UI design method. They did not introduce this method until the second year

> *"because it builds on other skills (drawing, sketching, thinking) and because it is fairly abstract, we cannot teach it earlier" (ibid, p75)*

They indicate that they have a comprehensive set of exercises for teaching the students how to write their own UI patterns. Unfortunately, in more general computer science and information technology curricula, UI design is normally taught in one or at most two courses so such an intensive, integrated four year approach is not practical. Koukouletsos describes a half semester course for introducing students to UI patterns focussed on usability of Web sites. This may be more appropriate, although considering the breadth of material required by most UI curricula this too may be considered too comprehensive a focus on UI patterns.

None of the studies discuss the UI design method they used with the students in any degree of depth and the resultant designs are mostly low or high-fidelity prototypes. Two of the more recent studies with professional subjects have involved conceptual model building (Lin & Landay 2008, Bernhaupt, *et al.* 2009). None of the education-oriented studies used UI patterns for guiding the development of any form of UI conceptual modelling, as described in the pattern-driven model-based UI design method described by Seffah and Gaffer (2007).

2.2.3 Potential for Learning and Teaching

Many of the findings from the teaching situation (Table 2.3) are similar to those described by researchers investigating pattern use by designers (Table 2.2). The studies by Wesson and Cowley (2003) and Kotzé *et al.* (2006) both report on the confusion new UI pattern users have in understanding the links between patterns. This similarity is not unexpected because there is often a significant overlap between the profiles of the research participants (e.g. novices) and the class composition of many UI courses. Findings from such research are likely to be indicators for using patterns in a teaching situation. For example the investigations carried out by Dearden *et al.* (2002) introduced UI related patterns to subjects who had a variety of backgrounds, from web design students to members of the public with no website experience.

The promised benefits of using patterns in UI design reported in the general UI patterns literature (e.g. Erickson 2000, Granlund *et al.* 2001) would appear to be similar for both UI designers and students although educational requirements provide a different focus for those benefits. Benefits can also be identified in other related literature including that concerning pedagogical patterns (Fincher & Utting, 2002, Sharp *et al.* 2003), introducing algorithmic patterns to students (Muller, 2005) and pattern-informed UI design methods (Seffah & Gaffer, 2007). Introducing students to UI patterns has the following potential benefits:

1. Help students process the knowledge more effectively because patterns organise UI design knowledge into chunks (Clancy & Linn 1999, van Biljon *et al.* 2004, Kotzé *et al.* 2008). Patterns encapsulate best-practice UI design and UI expert experience in an accessible form (Granlund *et al.* 2001, Sinnig *et al.* 2003).

2. Help communicate good design values to students (Borchers 2001a, 2001b, Fincher & Utting 2002, Mahemoff & Johnston 1998) and help them become more aware of users' needs (Barfield *et al.* 1994).

3. Help students to bridge the gap between abstract and physical representations (Wesson & Cowley 2003, Kotzé *et al.* 2008, Erickson, 2000, Seffah & Gaffer, 2007). A UI pattern language is effectively a meta-language from which project-

specific pattern languages can be produced (Erickson, 2000, Todd, 2004). Selected patterns representing a UI design can be instantiated to the design's domain.

4. Help students retain the UI design principles contained within UI patterns (Borchers 2002, Seffah 2003, Koukouletsos *et al.* 2009, Granlund *et al.* 2001, Kotzé *et al.* 2006).

5. Help students to develop better quality UI designs (Laakso *et al.* 2000, Griffiths & Pemberton 2004, Kotzé *et al.* 2006, Segerstahl & Jokela 2006, Wania 2008, Koukouletsos 2008, Mahemoff & Johnston 1998).

6. Help the average student more quickly learn and apply UI design principles (Laakso *et al.* 2000, Kolfschoten *et al.* 2010).

7. Help students develop UI designs for unfamiliar domains (Chung *et al.* 2004, Saponas *et al.* 2006, Koukouletsos *et al.* 2009).

8. Encourage a problem-oriented approach to UI design (Seffah 2003).

9. Encourage and enhance communication between students (Finlay *et al.* 2002, Chung *et al.* 2004, Griffiths & Pemberton 2004, Saponas *et al.* 2006, Schmidt *et al.* 1996, Mahemoff & Johnston 1998) by providing a common vocabulary – *lingua franca* that can be comprehended by all, regardless of background (Erickson 2000).

10. Help focus the discussion of students working in teams on relevant UI design problems (Davis 1986, Laakso *et al.* 2000, Bernhaupt *et al.* 2009). They may also help encourage consensus (Bernhaupt *et al.* 2009) and ensure all members of the team participate on an equal footing (Erickson 2000).

11. Aid student understanding of UI pattern content through the use of illustrated examples (Finlay *et al.* 2002, Chung *et al.* 2004, Segerstahl & Jokela 2006, Kotzé *et al.* 2006, Koukouletsos 2008).

12. Make the UI pattern content accessible to students through using a well defined and easy to read format (Dearden *et al.* 2002, Borchers 2001a, 2002, Koukouletsos 2008, Granlund *et al.* 2001).

13. Help students develop projects incrementally because patterns aid piecemeal UI development (Bayle *et al.* 1998, Erickson, 2000, Dearden and Finlay 2006, Schümmer and Slagter 2004).

14. Help students create quality documentation because patterns can enhance the documentation of completed UI designs (Goldfedder & Rising 1996, Mahemoff & Johnston 1998) and provide the background reasoning for design decisions (Granlund *et al.* 2001).

However, problems have been reported by researchers in both student and designer-oriented studies. Many of these problems were identified by the subjects themselves. Issues that need to be considered when introducing students to using UI patterns and UI pattern languages include:

1. Assimilating the information in UI patterns requires time to read and understand (Finlay *et al.* 2002, Chung *et al.* 2004, Segerstahl & Jokela 2006).
2. Excessive reliance can be placed on illustrative examples in the UI patterns without consulting the accompanying text (Dearden *et al.* 2002, Finlay *et al.* 2002, Koukouletsos *et al.* 2009).
3. A UI pattern's solution needs a visual representation to help students identify the essential elements (Chung *et al.* 2004).
4. Inappropriate pattern naming can lead to confusion and may result in inappropriate use of a UI pattern (Wesson & Cowley 2003, Kotzé *et al.* 2006, Segerstahl & Jokela 2006, Koukouletsos 2008).
5. Lack of understanding of the relationships linking patterns may lead to difficulty in locating additional required patterns (Dearden *et al.* 2002, Finlay *et al.* 2002, Wesson & Cowley 2003, Kotzé *et al.* 2006, Koukouletsos 2008).
6. When students are first introduced to using a UI pattern language a facilitator may be required to guide initial pattern selection using the links (Finlay *et al.* 2002).
7. The order and form of the patterns may influence students successful use of UI patterns (Dearden *et al.* 2002, van Biljon *et al.* 2004, Sharp *et al.* 2003).
8. The physical form in which the patterns are presented may influence how students use them (Dearden *et al.* 2002).

On balance, the potential advantages of introducing UI pattern languages to students seem to outweigh any potential problems. Educators need to ensure that issues identified above are addressed when introducing students to UI patterns and using a UI pattern language.

2.3 A Pattern Language for Teaching UI Design

To improve the learning experience for UI design students current pattern languages need to be further developed (Borchers 2002, Finlay *et al.* 2002). Koukouletsos (2008) extends this requirement by suggesting that a tailored UI pattern language based on existing UI pattern collections should be developed to improve the learning experience of students. He comments on the development of the pattern language he created for his students:

> *"Considerable effort is needed to develop a pattern collection or language acceptable and widely recognised as a rich body [...] providing advice and guidance." (ibid, p267)*

Three factors identified as influencing the development of such a teaching language that require further investigation are:

1. The nature of the information within the pattern, including the type of visual elements required (Laakso *et al.* 2000, Dearden *et al.* 2002a, 2002b, Finlay *et al.* 2002, Wesson & Cowley, 2003, Chung *et al.* 2004, Kotzé *et al.* 2006).

2. The structure and form of a pattern, i.e. the ordering of content and the physical presentation of each pattern to be used in student exercises (Dearden *et al.* 2002, Finlay *et al.* 2002, Sharp *et al.* 2003).

3. The organisation of the pattern language (Dearden *et al.* 2002, Wesson & Cowley, 2003b, Kotzé *et al.* 2006, Wania & Atwood 2009).

2.3.1 The Nature of the Information

The first of the factors identified as influencing the development of a teaching language focuses on pattern content because the reported studies imply that pattern content should be tailored to enhance student learning (Laakso *et al.* 2000, Dearden *et al.* 2002a, 2002b, Sharp *et al.* 2003, Chung *et al.* 2004, Kotzé *et al.* 2006, Koukouletsos 2008). A number of studies have identified that the visual elements of a pattern are important (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Chung *et al.* 2004, Segerstahl & Jokela 2006, Kotzé *et al.* 2006, Koukouletsos 2008). Finlay *et al.* (2002) reported that the three sections their subjects identified as useful were the problem statement, the illustrative examples and the solution précis. The illustrations were seen as the most helpful component. Observations by Chung *et al.* (2004) showed that the pattern names and diagrams illustrating the solutions were important, especially when their research participants were searching for a pattern to help them solve a problem.

The visual elements used in patterns could be very important when using patterns with students. Although in the reported studies the illustrations used in the patterns were mostly examples, sketches representing the essential elements of the solution were also present in some UI patterns. Both types of illustration are discussed in more depth in the following sections.

2.3.1.1 Illustrating Patterns

In the Alexandrian pattern form a sensitising image is a fundamental part of the pattern (Alexander *et al.* 1977). This pattern language (Alexander 1979) is made up of patterns that have at least two illustrations: the sensitising image after the name of the pattern and the sketch near the end of the pattern highlighting the significant elements of the solution. The sensitising image is essential to a pattern (Alexander 1979) and the large size, position and layout indicates its importance. The name and the sensitising picture are displayed on a single page with the remainder of the pattern's content on the following pages. Any illustrations accompanying examples in the body of the text are small (under 4x5cm). The solution sketch is about one third of a page. Size indicates that the sketch is more important than any illustrative examples.

In the UI domain Graham (2003) considers the sensitising image to be

> *"a picture or diagram concerning, supporting or illustrating the pattern" (ibid, p51).*

An examination of the many UI design-related pattern collections shows that there is a key image near the beginning of each pattern. There is generally no size difference between this illustration and those associated with the exemplars within the body of the pattern (van Duyne *et al.* 2003, Folmer *et al.* 2005, Snow *et al.* 2006, Tidwell 2006).

There are exceptions. The illustrations accompanying Borchers HCI collection (2001a) follow a similar size relationship as used in the Alexandrian patterns. Tidwell's Common Ground collection (1999) has no key illustration and examples are simply a list of the names of UI types. The comprehensive pattern collection of Schümmer and Lukosch (2008) has reversed the size difference of the Alexandrian patterns with a small half column key picture at the beginning of each pattern and larger images illustrating examples near the end of these patterns.

Many of the images illustrating the WU pattern language (Graham n.d.) are screen shots from existing websites, as seen in many other collections. But some are not, for example a photo of a spider at the middle of a web illustrates pattern 56 NATURAL METAPHORS. Similarly, the UbiComp collection (Chung, *et al.* accessed 2007) contains patterns where the images are from life, but these tend to be photographs including people. The key pictures of the Schümmer and Lukosch (2008) collection are also photographs, often of people. Some of the patterns in the Yahoo! Design library (Yahoo n.d.) are illustrated with animations.

When screen-shot illustrations dominate a UI pattern there is the problem that users may copy features shown in these illustrations before identifying the key features of a pattern to determine which are relevant to their design (Dearden *et al.* 2002b). Finlay *et al.* (2002) commented that:

> "*Some users focused exclusively on the illustrated examples.*" *(ibid, p10).*

Using alternative pattern structures and formats may be required in some situations to partially overcome this problem (Finlay *et al.* 2002, Sharp *et al.* 2003). The collection maintained by van Welie, originally known as the Amsterdam collection, has undergone major changes in structure over time and is now called "Patterns in Interaction Design". In one version of this collection (van Welie accessed 2007) the patterns are structured along the lines proposed by Dearden *et al.* (2002b). This updated version has the initial sections as: name, problem, solution précis followed by the key illustration. This layout may help focus a student's attention on the problem and solution and assist them to consider what aspects of any illustration are relevant.

Laakso (accessed 2007) in her teaching collection has also placed an explanation first but has not differentiated it into sections. This explanation is succinct and tries "*to emphasize the most interesting findings of each pattern*" (Laakso 2003). The illustrative examples that follow are clearly very important for these patterns. Each has an explanation of how the example instantiates the pattern, thereby helping students identify relevant features of each UI pattern.

Another approach to this problem can be seen in some of the UI patterns created by Tidwell (2006) and Laakso (n.d.) where a number of illustrations (e.g. Tidwell's pattern 39 DIAGONAL BALANCE and Laakso's RULE STORAGE) have been annotated to highlight key elements of the solution. A student using these patterns has their attention drawn to the significant features the exemplar illustrates. Although this approach does not overcome the problem of students simply using the feature in their own design without first considering its relevance, at least students would be copying a feature to which the pattern refers.

2.3.1.2 Solution as a Sketch

Given that sketching is a powerful tool for exploring UI design (Lin *et al.* 2002) the sketching method used in a teaching pattern language must be rich enough to guide students in the development of a model representing a user interface. Chung *et al.* (2004) observed that some of their patterns were not used whereas others were. They attributed this behaviour in part to the presence or absence of diagrams saying:

> *"hence visual representation of actual solutions worked best" (ibid, p238)*

Some UI pattern authors have chosen to provide an illustrative sketch identifying the main elements of the solutions for at least some of their patterns (e.g. Tidwell 1999, Constantine 2003b, de Paula & Barbosa 2003, van Duyne *et al.* 2003). The visualisation techniques used range from informal to formal: completely informal sketches (Tidwell 1999, 1998, Borchers 2001a), slightly less informal screen sketches and storyboards (van Duyne *et al.* 2003), semi-formal canonical abstract prototypes (Constantine 2003b) or wireframes (Bernhaupt *et al.* 2009) and formal MoLIC diagrams (de Paula & Barbosa 2003). As already discussed, such drawings could be as fundamental to UI design patterns as for architectural patterns, about which Alexander says "*a pattern must be drawable*" (Alexander 1979, p267).

For many patterns small free-hand sketches illustrating the solution can be both easy to create and to understand. On the other hand there is potential for them to be ambiguous to readers unfamiliar with symbols that the author may consider to be universally understood. In such situations there is no reliable way the reader can determine what the intended meaning might be. Such lack of standardisation and the potential for misinterpretation is undesirable for a teaching-focussed pattern language.

Visualisation of the solution provided via the use of sketching and storyboards, as in van Duyne *et al.*'s (2003) book, could be suitable because these techniques are accepted practice in user-centered design. They are regularly taught in introductory HCI courses so students will generally be familiar with interpreting them. Another advantage is that the sketching tool Denim (Lin *et al.* 2002) can be used to create these diagrams. Consequently, they can easily be included to illustrate new patterns and also to develop models of new UI designs. But sketches and storyboards also rely on the user knowing what the symbols represent, especially the widgets used to represent conceptual models of potential UI components. Many sketches tend to include elements that look like well known widgets. This can lead a student to visually matching to the closest widget in an actual tool box rather than considering the underlying concept the UI feature represents.

According to de Paula and Barbosa (2003), for many UI patterns the use of sketches to represent the solution is not sufficiently rich and what is required is:

> *"a representation language at a higher level of abstraction, and which focuses on the possible interactions users may have with the system" (ibid, p1).*

They propose the use of their Modelling Language for Interaction as Conversation (MoLIC) to represent UI interactions. They use MoLIC to represent the solution for patterns where simple interface visualisation is insufficient to capture the details of the interaction required in the design model. Using a formal approach like MoLIC overcomes the problem of not knowing how to interpret the symbol set. It also provides the students with a modelling technique that is removed from physical design elements. But it is a complex method with which students are unfamiliar. Experience indicates that students take time to learn how to effectively interpret and use more formal modelling systems. It would be hard to justify adding a new obscure technique to an already crowded curriculum. The other problem with MoLIC is that it focuses on interactions to the exclusion of other aspects of UI design. In a teaching pattern language all aspects of UI design need to be sketched, preferably with a consistent minimal symbol set.

The UI patterns Bernhaupt *et al.* (2009) developed relate to different stages of the development cycle in the UI design framework PD-MBUI. They use both semi-formal and formal methods for visualising the main components of a solution saying:

> *"According to the level of granularity, wireframes feature a schematic representation of the layout and the disposition of an UI element (e.g. a page or a form). Navigation diagrams are depicted using a formal description technique named StateWebCharts (SWC)." (ibid, p543)*

Wireframes are used for designing layouts which implicitly indicates that UI components have mostly been selected by this stage. Like sketching these focus students on toolbox components rather than the underlying concepts. But, like sketching, wireframes are relatively easy to create. SWC is a formal method and has similar difficulties to those identified for using MoLIC. This view is confirmed by

Bernhaupt *et al.* 2009) who reported that users with little or no knowledge of SWC did have problems.

For a teaching pattern language Constantine's canonical abstract prototypes (Constantine, 2003a) is a diagramming technique that can be easily sketched. Canonical abstract prototypes provide support for the creation of abstract or conceptual user interface models by students, based on the information contained in UI patterns. These could be sketched manually, or optionally semi-automated using software such as Interface Architect (Phillips & Joe, 2005) or CanonSketch (Campos & Nunes, 2004).

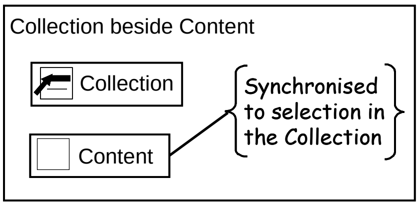


Figure 2.1- A simple abstract prototype example representing a collection of email messages with the content of the selected message displayed

Campos and Nunes (2004) show how simple interaction spaces can be used to illustrate spatial and layout features of a pattern solution. They use their own representation, Wisdom, for modelling navigational aspects of a pattern's solution rather than the navigation maps (Constantine 1998) that Interface Architect uses. A conceptual UI model is viewed as consisting of a collection of interaction spaces and dynamic linkages between those spaces. Interaction spaces are populated by abstract components (tools and materials) which are accessible to the user. Some spatial information may also be provided.

Canonical abstract prototyping is a semi-formal approach for creating conceptual UI designs. As can be seen in Figure 2.1, using these diagrams in a teaching pattern language has the advantage of being similar to sketching and storyboards. This minimises the learning hurdle for students, but has some of the advantages of a more formal approach. There is a clearly defined minimal symbol set so that students are focused on developing an understanding of UI design at a higher level of abstraction.

2.3.2 The Structure and Form of a Pattern

One of the major findings of the study by Finlay *et al.* (2002) was that the structure and format of patterns influenced their use. Borchers (2002) noted that he had to relax the restriction that students use the Alexandrian format for the patterns they were developing, and that students created well structured patterns. Unfortunately he does not indicate which format, if any, the students preferred. Presumably sufficient students had difficulty with the text-oriented Alexandrian form compared to the named section form used in the Common Ground collection (Tidwell 1999) to warrant lifting the restriction. On the other hand, when guiding the development of pedagogic patterns, Sharp *et al.* (2003) found that the participants preferred writing in Alexandrian form. Clearly pattern form is important and more research is required to discover the factors that influence the use of patterns in different domains, for different tasks and possibly the preferred learning style of the user.

UI design professionals and students both have problems using existing UI pattern collections. Research indicates that to improve usability requires a generalised, flexible pattern structure (Segerstahl & Jokela, 2006) for storing patterns. It is reasonable to conclude that a teaching language also needs a generalised and flexible pattern structure. This would enable pattern information including the type of visual elements to be selected and ordered for specific student exercises. Such a structure should enable pattern content to be presented in alternative forms for different audiences and for different situations (Fincher *et al.* 2002, Sharp *et al.* 2003). It should also help to identify common elements across collections, to recognize the diversity among patterns and to build thematic or meta-collections (Fincher, 2003b)

2.3.2.1 A Flexible Pattern Structure

A number of researchers recommended the use of XML to document patterns in order to make them more easily accessible from different software tools (van Welie & Traetteberg 2000, Borchers 2001b, Greene *et al.* 2002, Henninger *et al.* 2003). In 2003, the participants attending the CHI Workshop (Fincher, 2003a) specified the first version of a common form called PLML (Pattern Language Markup Language). It is an XML DTD that defines a template for a pattern with its elements and attributes. The development of PLML is an on-going project. An Extended Pattern Language Markup Language (PLMLx) was developed by Bienhaus (2004). An alternative development, PLML v1.2, was proposed by Deng *et al.* (2006). This individualises some fields so pattern content is more easily searchable. It identifies fields so that multiple examples with associated images can be defined. Changes extend the definition to include information that enables patterns from multiple collections to be searched and restructured in to new versions. This format also contains fields for maintaining a change log. This could be essential where educators are creating different versions of patterns for their students when completing different tasks. Kamthon (2006) and Henninger (2007) have described failings associated with using PLML. Henninger proposed using semantic net technologies to solve these problems and converted and extended PLML into an ontology. He defined a framework named PFOWL which enables patterns in different languages to be integrated via the inheritance structure. Kruschitz (2009) has proposed an alternative framework, XPML, with an emphasis on semantic mechanisms such as metadata, navigation and finding patterns. These frameworks rely on the use of W3C open standards such as XML, RDF and OWL.

An automated tool for managing collections of patterns structured using one of the derivatives of PLML could aid generation of existing patterns in alternative physical forms. Patterns could then be more easily tailored to fit students' needs as well as the nature of the exercises they are completing. It would be possible to review existing patterns that could then form the basis for constructing a teaching oriented pattern language.

2.3.3 The Organisation of a Teaching Pattern Language

This section focuses on identifying the kinds of relationships linking patterns together to form a pattern language. Indications are that the exploration of these relationships is important for understanding the underlying structure of pattern languages. Such knowledge is important for educators when selecting patterns for building a teaching-oriented pattern language.

2.3.3.1 Relationships

It is important to identify the type of links that exist between patterns because a unified pattern language for teaching requires an easily understood structure. In "The Timeless Way of Building", Alexander (1979) describes how a network of patterns can be built by describing the links between the patterns variously as '*contains*' or '*needs*' when referencing a lower level pattern. He describes the inverse relationship as '*contained by*' or '*is needed by*' which defines the context for a pattern (Alexander *et al.* 1977, px). He comments that the patterns have to be applied in a particular '*sequence*' to create an appropriate and complete representation.

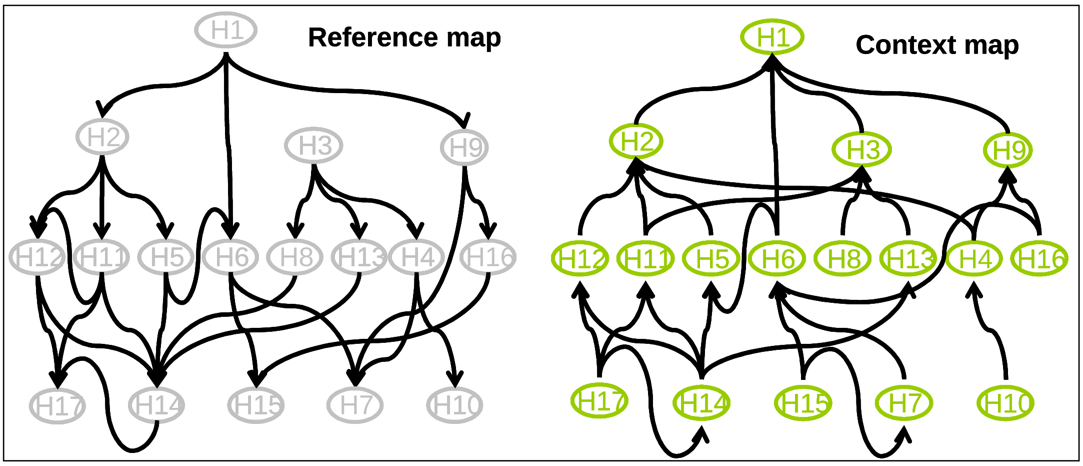


Figure 2.2 - Borchers' HCI language maps

Most patterns (except for those always at the apex of a hierarchy) complete others and need others for their own completion. Related patterns (those higher up in the hierarchy and those in the next level down) are mentioned in the context and reference sections of a pattern respectively. The context and reference map for the seventeen patterns (nodes H1 to H17) in Borchers' (2001a) HCI language are shown schematically in Figure 2.2. Note that the directional links do not accurately overlay in reality because it is difficult to obtain complete accuracy (Todd *et al.* 2003).

Alexander	Salingaros	Noble (Primary)	Noble (Secondary)	Fincher (PLML)	Schümmer
contains	contains	uses	tiling variant uses requires	contains	uses
contained by			used by	is-contained by	
generalise	derives	refines			
			refined by solution variant similar	is-a	specialises variant
sequence			sequence of elaboration		sequence
	complements		combines		
	overlapping and co-existence				
	alternatives	conflicts			similar

Table 2.4 - Examples of the variety of terms used to describe relationships

Another relationship type alluded to by Alexander (1979) is generalisation-specialisation. He comments that it may be possible to refine a higher level pattern from alternative solutions. Other types of relationships between patterns have also been identified. Table 2.4 compares the terminology used by selected authors.

When Salingaros (2000) subsequently investigated the nature of the connections in Alexander's pattern language (Alexander *et al.* 1977), he included '*contains*' and '*derives*' (for generalisation) but omitted the notion of sequence. He also identified another three relationship types: '*complements*, *overlaps*', and '*alternatives*'.

'*Complements'* is essentially the same as '*contains'* but the two patterns involved can appear at the same level rather than at different levels of the hierarchy (e.g. POSITIVE OUTDOOR SPACE and WINGS OF LIGHT). '*Overlapping coexistence'* refers to two patterns which solve distinct problems that overlap (e.g. HOUSE FOR A COUPLE and HOUSE FOR A SINGLE PERSON). Again, these patterns are all at the same level.

Alternative solutions to the same problem reflect the richness of a common language evolving from individual pattern languages, particularly if they act at different places in the hierarchy. According to Alexander (1979) there will be systematically different sets of connections between patterns e.g. one group of people might have a connection between COMMON AREAS AT THE HEART and FARMHOUSE KITCHEN whilst people in another neighbourhood do not.

To determine a full set of relationships, it is useful to consider Noble's (1998) classification, developed from analysing a corpus of software design patterns. He identified three primary and nine secondary relationships (Table 2.4). For the relationship types '*contains'* and '*generalise'* as identified by Alexander (1979), Noble (1998) uses the terms '*uses'* and '*refines*'. The principal secondary relationships '*used by* and '*refined by'* are the inverse of these. The third primary relationship type, '*conflicts'*, refers to patterns that provide mutually exclusive solutions to similar problems, like Salingaros' (2000) '*alternatives'*.

Three secondary relationships '*tiling'*, '*variant uses'* and '*requires'* are special cases of '*uses'*. On the other hand both '*solution variant'* and '*similar'* are instances of '*refined by'*. The former involves patterns where there are different tradeoffs between forces. The latter describes patterns with similar solution techniques. '*Sequence of elaboration'* refers, as in Alexander, to the list of patterns representing a specific design. When patterns that are not in a hierarchical relationship need to be linked to solve a problem, this is known as the '*combines'* relationship. This is similar to the '*complements'* relationship of Salingaros (2000).

Schümmer (2003b) in his article on groupware patterns mentions several kinds of relationships from which five key relationship types relevant to an HCI context have been selected. These names are quite different from those specified at the CHI 2003 workshop (Fincher 2003a) which defined three relationships – '*contains'*, '*contained by'* and '*is-a'* (specialisation). The issue of naming relationships is a problem not only in the HCI community but also for other communities in which pattern collections have been developed.

This discussion about relationships is not intended to be comprehensive, but sufficient to form a basis for discussing the issues that will arise when using UI pattern languages in an educational setting. The basic relationships that would be expected to occur in most UI pattern languages are those defined at the CHI 2003 workshop. A rich UI pattern language could also have examples of one or more of the other relationships. An important issue when teaching about user interface design, using patterns will be to determine which relationship types are the most useful and how large a set of patterns is appropriate for introducing students to UI pattern languages.

2.3.3.2 Tools for Managing UI Pattern Collections

Storing a teaching pattern language in a derivation of PLML should provide the required flexibility but this will not be easily utilised without a pattern management tool. Such a tool is required initially to provide easy access to existing UI pattern languages, then to enable ongoing modification as a teaching language develops. It should also enable subsets of patterns to be selected and output in a format suitable for specific exercises or groups of students.

A diverse group of pattern-oriented tools have been developed for creating, managing and using UI patterns. Deng *et al.* (2005) identified three categories:

1. Catalogue tools, which organise the patterns into different groupings for easy browsing and may also provide some assistance for the creation and submission of new patterns (e.g. Patterns in Interaction Design, van Welie www.welie.com/patterns/index.php).

2. Management tools, which provide facilities for customising and presenting the patterns in alternative forms as well as helping the user manipulate collections, provide templates for pattern creation and mechanisms for linking patterns (e.g. MUIP - Deng *et al.* 2005, MOUDIL Gaffer *et al.* 2003).

3. Pattern-based design tools, which provide support for pattern guided development of UI designs by automating some processes, such as, creating story boards or generating code (e.g. Damask - Lin & Landay 2008).

The functionality of the pattern-based design tools may be useful for students to access and use patterns when working on individual UI design exercises but UI design is primarily a team activity. As the design tools mature to fully functional groupware they may facilitate team design exercises. But a number of researchers (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Chung 2004) report that physical manipulation of patterns presented on separate sheets of paper was an important ingredient for stimulating quality discussion among members of design teams. More sophisticated UI devices such as interactive whiteboards may in future remove the need for using paper but currently none of the systems investigated has this type of functionality. Cataloguing tools could be useful for providing teachers and students with access to pattern collections but management tools provide the flexibility and functionality required for creating modified versions of patterns to create tailored pattern languages. With a management tool teachers should be able to create new teaching-oriented UI pattern languages based on existing collections while maintaining links back to the original patterns. They should also be able to create alternative ordering of content and different presentation formats. Based on the results of reported studies a fully functional management tool like the Managing User Interface Patterns (MUIP) tool (Deng *et al.* 2006) is probably the most useful type of tool for teachers of UI design.

MUIP (Deng *et al.* 2006) has facilities for identifying and creating relationships between patterns within and between different pattern collections. Being able to search and organise patterns based on different fields should aid exploration of pattern languages, help with comparison and help select suitable subsets of patterns for student use. As patterns can be exported in alternative formats this should enable tailoring of the physical presentation of individual patterns for use in student exercises.

2.4 Conclusion

This review focussed primarily on literature that can inform educators about using UI pattern languages as a teaching tool. A number of studies (Table 2.3) report experiences of using UI patterns with students. Some UI design studies (e.g. Dearden *et al.* 2002b, Chung *et al.* 2004) include either student participants or novice designers. All of the studies identified a number of advantages in using patterns, with those involving teams reporting that some aspect of communication between team members was enhanced. Chung *et al.* (2004) reported that prior UI design experience was more important than using patterns when evaluating the designs created during their experiments but patterns did assist less experienced members. Similarly, Laakso noted that providing a set of patterns targeting UI design errors commonly made by students, helped lower-performing students complete exercises more quickly and produce better quality designs. Importantly, Borchers (2001b) reported that the students found patterns useful.

Thirteen benefits of using UI patterns and pattern languages in an education setting have been identified. Some of these concern patterns or pattern languages specifically, for example making the material accessible through a well-defined format and at a suitable level of granularity. Others relate to process, encouraging a problem-oriented approach to design, helping students retain UI design principles, and assisting them to bridge the gap between abstract and physical UI representations. In addition, there is the promotion of design values and helping students to become aware of the needs of others in order to develop solutions which have the ‘quality without a name’.

The studies reviewed identify issues that may cause problems when introducing students to UI patterns and pattern languages, and eight of these have been identified. Two examples being Finlay *et al.* (2002) and Chung *et al.* (2004) had feed-back from subjects that patterns contained too much information to assimilate within the time available. Some of the studies (Dearden *et al.* 2002b, Wesson & Cowley 2003, Kotzé *et al.* 2006) reported that their subjects had problems following the relationships between patterns and hence developing an understanding the structure of a pattern language.

Often relatively simple issues proved to be significant, for example Wesson and Cowley (2003) and Kotzé *et al.* (2006) note that pattern names could mislead a designer trying to identify useful patterns. Even the order of the sections in a pattern could influence use (Dearden *et al.* 2002b, van Biljon *et al.* 2004) as could the physical form of the pattern (Dearden *et al.* 2002a, 2002b).

There is clearly a need for a standardised pattern form (Fincher 2003a) such as using a derivation of PLML as the basis for the storage structure for patterns. Such a form should facilitate easier selection and manipulation of content, and presentation of that content in a form preferred by the user. Such a form would facilitate further research into the selection and ordering of the UI knowledge contained in pattern form for students. Investigations are needed to identify those sections of UI patterns that students find most useful, and both how and where to define the relationships linking patterns into a pattern language.

In using UI patterns the intention is not to introduce students to a corpus of solutions but to assist them in understanding the underlying UI design principles. To attain this aim both the structure of the individual patterns and the pattern language used, need to be tailored specifically to the needs of the students. Although three factors were identified from the literature as being significant to student learning there appears to be no research that tested whether these factors influence student acceptance and the usefulness of patterns as an aid for teaching about UI design. In the context of teaching UI design, further research needs to identify:

1. How best to demonstrate the organisation and structure of a pattern language.
2. Which components of a pattern students find most useful and the significance of:
    - Pattern names and the vocabulary used within a pattern,
    - Pattern content order,
    - Illustrations in patterns, including sketches highlighting a pattern’s solution.
3. What tools and techniques can successfully introduce students to using a pattern language.

Chapter 3: Characterising UI Pattern Languages

Mullet (2002) has identified the organisation of a pattern collection as a major problem when using patterns to guide UI development. This problem has been described by other users of patterns when designing new user interfaces (Wania 2008, Finlay *et al.* 2002, Dearden *et al.* 2002a, b, Wesson & Cowley 2003). A similar problem has been observed when introducing UI design students to working with a collection of patterns (Dearden *et al.* 2002a, b, Wesson & Cowley 2003, Kotzé *et al.* 2006, Koukouletsos 2008).

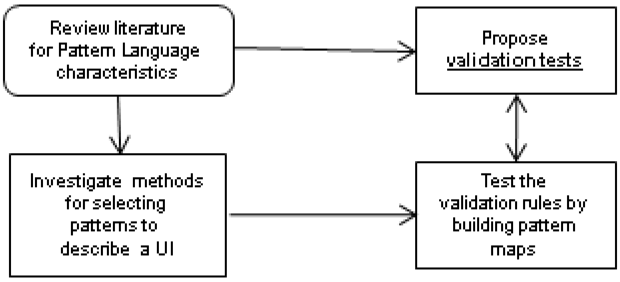


Figure 3.1 - Process followed for this investigation

The participants in the CHI2002 UI patterns workshop (McInerney 2002) concluded that pattern language organisation was an issue that required addressing, suggesting one possible cause as being:

> *"Collections lack guidance on how to use patterns together as components to solving a larger design problem." (ibid, p3)*

This chapter addresses issues related to the problem of recognising and using UI pattern language structure. Figure 3.1 shows the inter-relationships between the different parts of this investigation.

After presenting a number of pattern language definitions as background, the literature associated with validating pattern languages is discussed. Next, methods for selecting patterns to model existing user interfaces are investigated. A set of validation and maturity rules for determining the characteristics of a well structured UI pattern language is proposed.

3.1 UI Pattern Languages

Patterns use and are used by other patterns forming a pattern language. Alexander (1979) observes that a pattern language "*not only connects the patterns to each other but helps them come to life*" (ibid, p315). There is a notion of hierarchy, with the patterns below a higher level pattern referred to as its principal components.

Nonetheless this hierarchy includes networks, and networks of networks. There is also the concept of scale with smaller patterns being lower down in the hierarchy. A good pattern language, therefore, depends on the structure of the network linking the patterns together. A language is complete both morphologically and functionally when it can be used to visualise the details of a solution and resolves all relevant forces (e.g. a language for building houses). From a set of individual pattern languages, a larger, common language can evolve (Alexander 1979).

There are many definitions of UI pattern languages but one by Borchers (2001a) captures the main characteristics:

> *"A pattern language is a hierarchy of design patterns ordered by their scope. High-level patterns address large-scale design issues and reference lower level patterns to describe their solution."* *(ibid, p8)*

Based on this definition and the preceding discussion, it is clear that a collection of related patterns that are organised and linked into one or more interlocking hierarchies may be referred to as a pattern language. However, it is not clear when a collection of patterns evolves into a pattern language. Appleton (1997) provided a detailed discussion of pattern-related terminology as it was used in the early stages of UI pattern development. He defined a pattern language as:

> *"… a collection of such solutions [patterns] which at every level of scale, work together to resolve a complex problem into an orderly solution according to a predefined goal."* *(p 16)*

This definition like that of Borchers (2001a) highlights the underlying hierarchical nature of a pattern language based on some identifiable scale factors. It describes the concept of the shared objective or predefined goal which could identify a root for a pattern map. This definition indicates that the links between the patterns should exhibit some recognisable harmony. In an analysis of the Alexandrian pattern language (Alexander *et al.* 1977) Salingaros (2000) also highlights these "*inherent structures and relationships*". He considers that it is the connectivity rules between patterns that make a collection of patterns into a language:

> *"A pattern is an encapsulation of forces; a general solution to a problem. The 'language' combines the nodes [patterns] together into an organizational framework."* *(ibid, p154)*

Pemberton (2000), when discussing UI pattern languages again identifies connectivity between patterns as the attribute that adds power to a pattern language:

> *"The fact that individual patterns are integrated into pattern languages … enables the collection of patterns to operate generatively, each pattern showing the sub-patterns required to resolve more detailed design issues, in the context of the larger design." (ibid, p5)*

Borchers (2001a) agrees with Pemberton (2000) and Salingaros (2000) and also provides a more formal definition that identifies the synergistic effect when using patterns together rather than individually saying:

> *"The context, together with the references, represents the added value that turns a loose collection of patterns into a pattern language." (Borchers 2001a, p71)*

1. A pattern language is a directed acyclic graph PL= (P, R) with nodes P = [P1,..,Pn] and edges R = [R1,...Rm].
2. Each node $P \in P$ represents a pattern.
3. For two patterns P and Q $\in$ P it is said that P reference Q if and only if there is a directed edge R $\in$ R leading from P to Q.
4. The set of edges coming from a pattern P $\in$ P is called its references and the set of edges coming to a pattern is called its environment.
5. Each pattern P $\in$ P is a n-uple:
   P = (n, a, c, co, p, s, ctx, f1,...fn, e1,.ej, ce1,...cem, p1,...pn) where:

n name	a author	c classification
co confirmability	p problem	s solution
ctx context	f1,..,fi forces	u usability
con consequences	e1,..,ej examples	ce1,..,cem counterexamples
p1,..,pn related patterns		

Table 3.1 - A formal pattern language definition (Acosta & Zambrano 2004, p80)

Borchers (2001a) also provides a formalised definition of a pattern language that emphasizes connectivity. It was modified by Acosta and Zambrano (2004) primarily by extending the element list that defines an individual pattern's content and by using the term environment rather than context when defining the directional links between patterns. Table 3.1 shows their formal pattern language definition.

The different definitions all indicate that the links between patterns on one level can form a higher-level pattern that includes information not available from the individual patterns alone. This additional information is not available in the constituent patterns at the lower level. Grouping that links lower-level patterns to higher-level patterns creates a hierarchy of scale. This implies that it should be possible to describe a user interface at different levels within a UI pattern language hierarchy. A UI can be described at different levels of granularity, just as a road map can be either high level showing just the main highways, or low level showing details of the transport network from highways down to walking tracks.

Salingaros (2000) makes the comment that:

> *"A loose collection of patterns is not a system, because it lacks connections" (ibid, p154)*

Thus implying that the quality and nature of the connections between patterns is what determines whether a collection is a pattern language or not.

3.2 Validating UI Pattern Languages

Only a few studies seem to have considered the quality of a pattern language's structure (Borchers 2001, Fincher 1999, Salingaros 2000, Todd *et al.* 2003, 2004). Salingaros (2000) identifies two forms of connectivity when discussing pattern languages: external connectivity and internal connectivity. These two forms of connection are central to validating a pattern language:

External validation - Considers the relationship the language has to human function or behaviour, or the "feel right" factor.

Internal validation - Examines the connectivity between the levels in the language's hierarchies to determine the "*ability to combine*" to describe higher order patterns.

External validation is related to the "*Value System*" identified by Fincher (1999), which refers to that attribute of a pattern language that "*is reflected by, and embodied in, their sense of audience"* (ibid, p2). It relates to Alexander's '*quality without a name*'. For user interfaces there are two obvious audiences, UI design professionals and the users who will work with the resulting system. A third group is students or trainees of UI design. According to Salingaros (2000) if a pattern language has external validity then the reader should recognise this if they:

- Approach the language from the bottom-up,
- Can progressively build up in their mind the connectivity map from the small to the large in a natural progression,
- Feel a sense of connection with the lower order patterns, because these patterns relate to their own experience.

This human dimension is an attribute of Alexander's pattern language (Alexander et al. 1977). , When discussing external validation Salingaros (2000) highlights this with the following observation:

> *"… what demonstrates the patterns' inevitability is their connection to fundamental patterns of human behaviour and movement" (ibid, p153)*

When discussing external validity Salingaros (2000) implies that the pattern language first has to have internal validity. Interacting with a computer via a user interface is a human activity and the association between behaviour and action identified within the architectural domain may also be present within the UI domain. In more pragmatic terms external validation can be discussed in the terms identified by the CHI2002 workshop participants (McInerney, 2002) as suitable for evaluating a pattern language: breadth, depth, applicability, clarity and convenience.

Using a UI pattern language that has proven external validity to guide UI design is more likely to result in a UI design that possesses the '*quality without a name'*.

> *"Interaction designers can also aim at the quality without a name. When they succeed, their interfaces become "transparent" and the interaction acquires a natural feel - it becomes a natural element of a user's daily life." (Barfield et al. 1994, p71)*

Measuring such value is highly subjective. But other researchers do report improvements in the quality of the interfaces students design when they are encouraged to use UI patterns to inform their design efforts (Borchers 2001a, 2002, Griffith & Pemberton 2004, Koukouletsos *et al.* 2009, Laakso *et al.* 2000).

Internal validation examines how related patterns are linked together. A graph created by connecting every pattern in a pattern language is referred to as the language map. Salingaros (2000) makes the observation that a pattern language map is not a simple hierarchical tree structure because a pattern language may have more than one root. Although he implies a sub-system within the language may culminate in a single root node.

When discussing what characterises patterns and pattern languages, Fincher (1999) identified five elements that need to be considered. Of these only the '*Organising Principle*' relates to internal validation. It refers to the way patterns relate to each other, so that they can be arranged based on some recognisable scaling factor such as system level to widget level. Salingaros (2000) agrees that this principle is important to internal validation saying:

> *"One of the principal methods of validating a pattern language is that every pattern be connected vertically to patterns on both higher and lower levels." (ibid, p156)*

When defining a pattern language, Borchers (2001) identifies the context of a pattern as those patterns that reference it (environment in Table 3.1). This indicates that pattern languages which comply with Borchers definition should have the same map for the context links as for the reference links. He says "*the context is the 'inverse function' of the references*" (ibid, p72). This observation is consistent with research by Salingaros (2000), which shows that internal validity of a pattern language can be established by examining the connectivity between patterns. He says:

> *"Graph theory visually illustrates some key aspects of pattern languages: how patterns combine to form higher-level patterns containing new information; how linked patterns exist on different levels; how to find patterns in a new language; and how a pattern language is validated through its connective structure independently of each individual pattern's validity" (ibid, p149)*

The Alexandrian pattern language (Alexander *et al.* 1977) has different context and reference maps, which Borchers (2001) explains is because:

> *"Alexandrian patterns are, above all, a didactic medium for human readers, even (and especially) for non-architects. To Alexander this quality has priority over a mathematically correct representation" (ibid, p22)*

Salingaros (2000) does not draw attention to the inconsistencies in the Alexandrian pattern language but he does indicate that languages are developing and evolving, and may contain inconsistencies at any point in time. He discusses how pattern languages may evolve indicating that as a pattern language matures the connections within the levels increase and a pattern language:

> *"… develops coherence over time [and] may also develop a degree of self-similar scaling as a result of the connections across levels" (ibid, p159)*

Salingaros implies that the connections between levels and within levels in a pattern language are factors in determining a pattern language's maturity. By developing the map of an existing UI pattern collection the UI designer should get an indication of the maturity status of the collection. It should be possible to organise nodes within the map into one or more hierarchies where the higher-level patterns provide a conceptual description of an interface and also provide the context in which the lower level patterns could be used. Reading across levels could provide descriptions at different degrees of granularity.

Both Salingaros (2000) and Borchers (2001) identify spatial hierarchies based on size from small-scale objects up to larger-scale ones. Borchers says that there are two spatial dimensions, possibly three, that need to be considered in UI pattern language hierarchies. He does not identify them. Both writers identify a temporal hierarchy that relates patterns that follow each other sequentially in time (i.e. an object based on one pattern cannot be accessed before an object based on a proceeding pattern has been accessed). Each of the different types of hierarchy may provide an alternative view of participating patterns. These provide alternative views of the user interface, like the different views of real world features provided by: topographical maps, cadastral maps and marine charts.

3.3 Selecting UI patterns from a Collection

To develop a better understanding of the internal structure of UI pattern languages two collections that at the time had featured in other studies (Borchers 2001, 2002, Dearden et al. 2002a, b, Laakso *et al.* 2000, Wesson & Cowley 2003) have been used to guide the development of descriptions for existing UIs. From the literature a number of different UI design methods that used patterns have been identified but most (Section 2.1.3) were not specific about the procedure for selecting patterns from an existing set of UI patterns. At the time this study took place two of the more complete descriptions were those by Tidwell (1998, 1999) and van Duyne *et al.* (2003).

One method for selecting patterns both these authors suggested is to simply pick out patterns that appeared useful from the collection. Tidwell (1999) suggested

> *"... read through the language, and pick out the patterns" (ibid, p9)*

This approach does not use any of the relationships between patterns to guide the selection process. To explore the organisational features of pattern collections, more formal processes have been investigated for selecting patterns to 'model' existing UIs.

3.3.1 The Common Ground Pattern Collection

Tidwell in "Common Ground" (1999) suggested two ways her collection of patterns might be used.

QUESTIONS
1. What is the basic shape of the **content**?
2. What is the basic shape of the **actions** taken with the artifact?
3. How do the content or available actions **unfold** before the user?
4. How does the artifact generally use space and the user's **attention**?
5. How is the content or action **organized** into working surfaces?
6. How can the user **navigate** through the artifact?
7. What specific **actions** should the **user** take?
8. How can the user **modify** the artifact?
9. How can the artifact be made **visually clear** and attractive?
10. How else can the artifact actively **support** the **user**?

Table 3.2 - Tidwell's (1998) set of UI design guidance questions

Tidwell's first approach uses a series of questions to guide pattern selection. The questions are reproduced in Table 3.2 with the key words highlighted in bold. Each question is associated with a group of patterns which provide solutions to different parts of the problem identified by that question. The suggested approach starts with "*broad strokes*" by selecting patterns that answer the first two questions about the UI being modelled. The approach suggests working down through the questions, selecting patterns for each one, to describe the UI in increasing detail.

SUB-LANGUAGES
Narrative
High-density Information Display
Status Display
Control Panel
Form
WYSIWYG Editor
Composed Command
Social Space
Navigable Spaces
Step-by-Step Instructions

Table 3.3 - Tidwell's (1999) set of UI design sub-languages

Tidwell's second approach uses sub-languages, an alternative organisation of the patterns. These sub-languages are listed in Table 3.3. Each sub-language groups together patterns that are regularly used in combination with each other and are based on one dominant pattern which gives its name to the sub-language. When a designer identifies one of these sub-languages as appropriate for their design they can restrict selection of patterns for describing details to just the patterns contained within the sub-language.

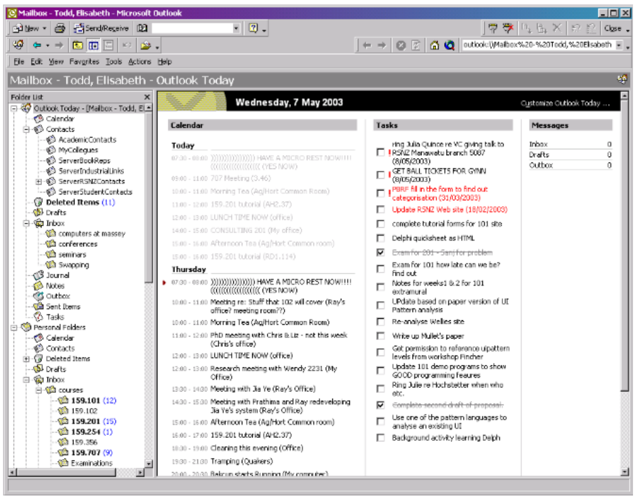


Figure 3.2 – Screen dump of the Outlook Today view, 2003.

To test these two approaches, patterns have been selected to describe the 'Outlook Today' view from Microsoft's Outlook tool (Figure 3.2). This interface has been chosen because it is comparatively well known, widely used and relatively complex. Applying Tidwell's first approach, twenty-three patterns have been selected to describe the example Outlook Today view and these are listed in the PATTERNS column of Table 3.4 (col. 2).

Using Tidwell's second approach, two sub-languages have been selected from the list of sub-languages shown in Table 3.3. The set of patterns associated with these two sub-languages are listed in the SUB-LANGUAGE columns of Table 3.4 (cols 3 & 4). Eight of the eleven patterns in the 'High-density Information Display' sub-language (col. 3), are relevant (non-relevant patterns *italicised*). Nine of the twelve patterns from the 'Status Display' sub language (col. 4) are relevant. The two selection approaches resulted in the identification of many common patterns. But, the sub-language-based approach identified patterns from only five of the ten groups identified by the question-based approach.

QUESTIONS	PATTERNS	High-density Information Display SUB-LANGAUAGE	Status Display SUB-LANGAUAGE
1 What is the basic shape of the **content**?	2 High density info display 3 Status display	2 High density info display	3 Status display
2 What is the basic shape of the **actions** taken with the artifact?			
3 How do the content or available actions **unfold** before the user?	8 Navigable spaces 9 Overview by detail 11 Small groups of related things 12 Hierarchical set, 15 Optional detail on demand 16 Disable irrelevant things 17 Pointer shows affordance 18 Short description	8 Navigable Spaces 11 Small Groups of Related Things *? Series of Small Multiples* 12 Hierarchical Set *13 Tabular Set* *14 Chart or Graph* 15 Optional Detail On Demand 16 Disabled Irrelevant Things 18 Short Description	11 Small Groups of Related Things 12 Hierarchical Set *13 Tabular Set* *14 Chart or Graph* 15 Optional Detail On Demand 16 Disabled Irrelevant Things 18 Short Description
4 How does the artifact generally use space and the user's **attention**?	19 Sovereign posture		
5 How is the content or action **organized** into working surfaces?	23 Tiled working surfaces		
6 How can the user **navigate** through the artifact?	26 Map of navigable spaces 27 Clear entry points		
7 What specific **actions** should the **user** take?	31 Convenient environment action 34 Choice from a small set 35 Choice from a large set 40 Toolbox		34 Choice from a Small Set *36 Sliding Scale*
8 How can the user **modify** the artifact?	41 User preferences 42 Personal object spaces	44 User's Annotations	41 User Preferences 42 Personal Object Space
9 How can the artifact be made **visually clear** and attractive?			
10 How else can the artifact actively **support** the **user**?	48 Remembered state 51 Important message		51 Important Message

Table 3.4 - Selected patterns describing the Outlook Today view using questions and sublanguages from Tidwell's collection (1998)

The sub-language-based approach to selecting patterns results in an incomplete list of patterns to describe the Outlook Today example. Therefore this approach would only be a starting point for selecting patterns to describe a UI. Following the question-based approach results in a more complete list of patterns. Attempting to describe the Outlook Today example by following the links is difficult, confirming the views of other researchers (Dearden *et al.* 2002a, b).

3.3.1.1 The Pattern Map

Tidwell notes that her collection of patterns does not appear to be in the form of an Alexandrian pattern language because as she says:

> *"I haven't yet been able to draw a coherent diagram of the whole language, nor define clear linear paths through it."* *(Tidwell 1999, p9)*

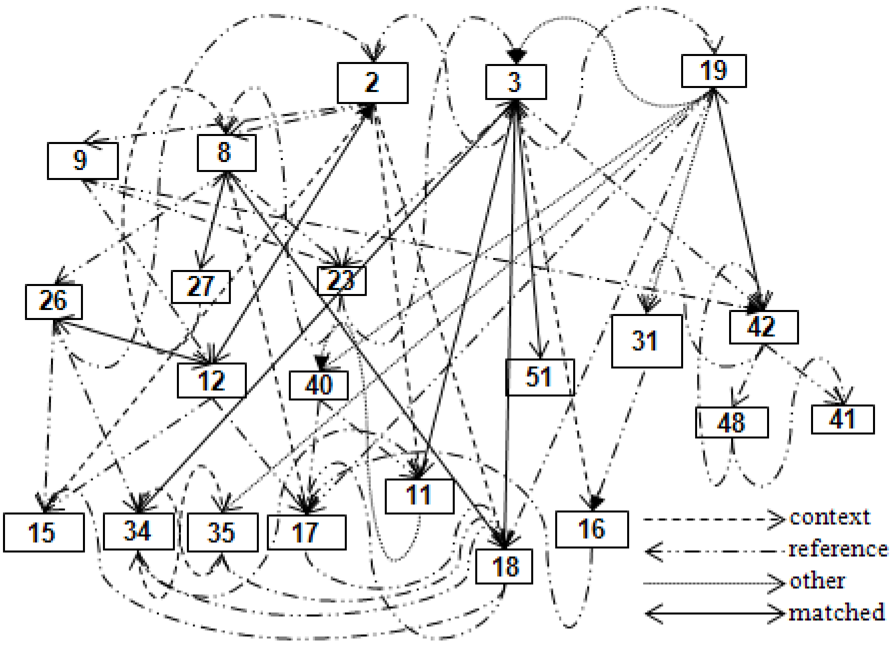


Figure 3.3 – Pattern map of the twenty-two linked patterns describing Outlook's 'Outlook Today' view

To check this assertion a graph has been created linking just the twenty-two patterns selected using the question-based approach, as shown in Figure 3.3. This graph (called a pattern map) identifies every link mentioned in any section of the participant patterns. The links have been classified into four groups.

The majority of the links can be found in either the 'Resulting context' section or the associated 'Notes' section of the patterns comprising the pattern map. These are references to patterns that could be used for designing further details of the pattern that contains the reference. These links are classified as '*reference*' links (similar to PLML's '*contains*' links). An example is the link between patterns 12 and 15.

The next two groups of links identify those patterns mentioned in the context section of a pattern (similar to PLML's '*is-contained-by*' links). Where the link in the context section of pattern A identifies pattern B, and pattern B contains a reference link back to pattern A, then the links are combined and classified as '*matched*'. An example is the link between patterns 8 and 27. Where there is no match then the links are classified as '*context*' links. An example is the link between patterns 2 and 11.

The last group of links are those that are found in the solution section of a pattern. An example is the link between patterns 19 and 3. These are classified as '*other*'.

Analysis of the graph seems to support Tidwell's assertion that the graph structure for the Common Ground patterns is incoherent. Only nine of the context and reference links between pattern pairs match each other. The link structure appears to be overly complex. There does not appear to be any obvious structuring into one or more interlocking hierarchies and levels of scale are not easily identifiable.

3.3.2 The Design of Sites Pattern Collection

Like Tidwell (1999), van Duyne *et al.* (2003) place their suggestions for pattern selection in the context of user-centered design. One major difference is that Tidwell's patterns are for general interface design while van Duyne *et al.* specifically target their patterns at web design. van Duyne *et al.* classified their patterns based on the types of issues a UI designer needs to address. They used statements to identify the twelve groupings from A to L as shown in the Id column of Table 3.5 (Col. 1).

Two approaches have been by suggested van Duyne *et al.* (2003) for selecting suitable patterns based on the organisational features of the collection. The first approach is similar to Tidwell's question-based one, in that it advises the designer to start by selecting a pattern from the Site Genres set, then move through the groups from A down to L selecting appropriate patterns from each group.

Their second approach uses the patterns listed in the section labelled "CONSIDER THESE OTHER PATTERNS" at the end of each pattern. This section corresponds to the section in the Common Ground patterns (Tidwell 1999) labelled 'Resulting context'. In the Design of Sites patterns the equivalent to Tidwell's 'Context' section is labelled BACKGROUND. These lists define the link structure of the Design of Sites pattern language.

ID	NAME	BACKGROUND (context)	CONSIDER THESE OTHER PATTERNS (reference)
A - SITE GENRES			
A8	Educational Forums	A2, A3	A9, B1, C1, ***D2***, D4, D5, E2, E6, H1, H2
B - CREATING A NAVIGATION FRAMEWORK			
B1	Multiple Ways To Navigate	A1-11	B2, B9, H1, H8, I6, J1, J2, K2, K4, K9
B2	Browsable Content	B1	B3, B4, B5, B6, B7, B8, B9, ***D2***, E1, I1, I3, K2, K3, K6
B3	Hierarchical Organisation	B1, B2	B4, B5, B6, B7, B8, K11
B4	Task-Based Organisation	B1, B2	B3, B8, H1
B8	Category Pages	B1-7	D1, ***D2***, I1, I3, J1, K2, K3, K6
C- CREATING A POWERFUL HOMEPAGE			
C1	Home Portal	B1	B1, B2, C2, D1, ***D2***, D4, E1, E4, I1, I2, I3, K2, K4, K7, K9, K10, L2, L3
C2	Up-Front Value Proposition	C1	E1, I3
D - WRITING AND MANAGING CONTENT			
D1	Page Templates	A1-11, B2, B8	***D2***, I1, I3, I4, I5, I6, K2, L1, L2, L,3 L5
D2	*Content Modules*	B2, D1	D1, D3, D4, **I6**, **K8**
D3	Headlines And Blurbs	A2, A3, A11, C1, D1, ***D2***, D5	D7, D9, I6
D7	Inverse-Pyramid Writing Style	B1, B2, B3, B9	D3, D6, D9, I2, K7, K9, K11
E – BUILDING TRUST AND CREDIBILITY			
E1	Site Branding	A1-11, C1, C2	I3, K2, L1
E6	Secure Connections	E3	E3, F1, K4, H1, H2, H4
F – BASIC e-COMMERCE			
G - ADVANCED e-COMMERCE			
H – HELPING CUSTOMERS COMPLETE TASKS			
H1	Process Funnel	A1, A4, A10, A11	A1, A4, A10, A11, E1, F1, G4, H2, H5, H6, H7, H8, I2, K2, K3, K4, K5, K6, K7, K12, K13
H2	Sign-In/New Account	D4, G3	D6, E3, E4, E6, H1, K12, K13
I – DESIGNING EFFECTIVE PAGE LAYOUTS			
I1	Grid Layout	D1, I2, I3, I4, I5	D1, ***D2***, I2, I3, J1, K2, L3
I3	Clear First Reads	D1, I1, I2, I4, I5	D1, I1, I2, I4, I5, K2, L2, L3
I5	Fixed-Width Screen Size	D1, I2, I3	D1, I1, I2, I4
I6	Consistent Sidebars Of Related Content	B2, D3, G2, K1	D1, ***D2***, D3, D4, D8, G2, K1, K7, K8
J – MAKING SITE SEARCH FAST AND RELEVANT			
J1	Search Action Module	B1, B2	I2, K4, J2, J3
J2	Straightforward Search Forms	J1	J1, J3
J3	Organised Search Results	J1, J2	B9, D6, D9
K– MAKE NAVIGATION EASY			
K2	Navigation Bar	B1, B2, J1	I3, K3, ,K10, L2, L4
K4	Action Buttons	B1, H1	K2, K5, L1, L2, B9
K7	Embedded Links	B1, B2, K10	K10, K9, K8, H6
K8	External Links	A6	K9, ***D2***, H1, H6
K9	Descriptive, Longer Link Names	B1, B2, I6, K2, K8	B9, D7, K7, K10, K11
K10	Obvious Links	K2, K6, K7, K8	B9, K2, K4, K6, K7, K8, K9, K11
K11	Familiar Language	B2, I3, D7	B2, K2, K9, K10
K12	Preventing Errors	K13	F1, K13, H1, H2
K13	Meaningful Error Messages	K12	F1, H1, H2, K11, K12
L – SPEEDING UP YOUR SITE			

Table 3.5 - Patterns selected to describe parts of Massey University's web site (2003)

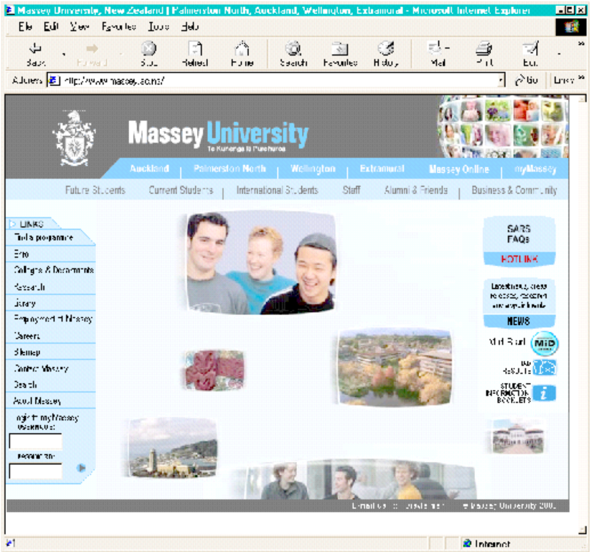


Figure 3.4 - Screen dump of the Massey University's Home page (accessed April 2003)

To trial the two approaches, patterns have been selected to describe the home page of Massey University's web site, Figure 3.4. This was typical of university sites at that time, and this was a site familiar to all students at the university. First, the groupings approach defined by van Duyne *et al.* was applied. Starting with the site genre pattern A8 EDUCATIONAL FORUMS, patterns from each relevant group have been selected from the ID column in Table 3.5 (col. 1).

When selecting patterns a broad interpretation of the patterns had to be used. Some of these patterns not only describe the UI design details but also refer to software design and implementation details. An example is when using the patterns to model an existing site from a user's perspective as there is no way of knowing whether the site is dynamic with content stored in a database. Patterns with such references have been included because the visible features and behaviours appear to conform to the pattern's descriptions. The thirty-two selected patterns are listed in the NAME column of Table 3.5 (col. 2). They are organised by pattern group.

To trial van Duyne *et al.'s* second linked-based approach, a pattern map was constructed but it proved to have too many patterns and links to be helpful. An equivalent tabular representation was then created but this too was difficult to navigate. Therefore a modified method has been developed that uses the subset of the patterns selected using the grouped-based approach. Using the tabular representation, for each pattern identified in the NAMES column of Table 3.5, the context and reference links have been listed in the BACKGROUND column and 'CONSIDER THESE OTHER PATTERNS' column shown in Table 3.5 (cols 3 & 4).

Analysis of these links shows much repetition in the lists, for example the pattern D2 CONTENT MODULES is referenced from eight different patterns from six of the groupings (Bold). D2 also references five patterns of which four have been selected for describing some aspect of the Massey University home page. Of these four patterns, only one (D3) has a matching reference in the Background section (context). The other three referenced patterns (D1, I6 & K8) also re-reference pattern D2 effectively creating a set of circular references.

Inheritance does not appear to have been used in structuring this collection of patterns. If the pattern D2 is implemented to manage the active areas on pages in the web site it seems logical it would be implemented for the whole web site. Therefore it only needs to be associated with the first pattern in the network A8 EDUCATIONAL FORUMS. But, the multiple references to patterns should make independent use of a pattern easier.

There appears to be a tension between defining links so that a pattern can be a standalone item of UI knowledge and defining links so the synergistic power of a pattern language can be realised. Defining patterns for standalone use results in an overly complex web of redundant links. This complex network would not encourage students or novice to use the link structure because many more decisions have to be made than would be necessary if the pattern languages' structure was simplified by removing redundant links. This conclusion, leads into the research question guiding the next stage of this investigation - what attributes characterise a well structured UI pattern language?

3.4 Measuring Internal Validity

It is not clear when the structure of a collection of patterns has developed to the extent that it should be called a pattern language. One approach that might make this point clearer is to evaluate a pattern collection's internal validity. Based on the earlier literature review the following tests (Todd *et al.* 2003) have been proposed to identify the attributes of a pattern language that contribute to internal validity:

Test-one: Do the *reference* and *context* links between the patterns form a pattern map? (*Borchers 2001, Fincher 1999, Salingaros 2000*)

A *reference* relationship links into those 'lower-level' patterns needed to instantiate the pattern. The *context* relationship defines a 'parent' pattern or part of the context in which the pattern can be used.

Test-two: Can the map be ordered into a hierarchy of layers? *(Borchers 2001, Fincher 1999, Salingaros 2000)*

Once one or more roots have been identified then the number of levels below a root can be counted. A level is counted if it includes more than one pattern. A sequence of patterns does not necessarily indicate the same number of levels.

Test-three: Can the layers making up the hierarchy be used to describe a UI at different levels of granularity? *(Fincher 1999, Salingaros 2000)*

This criterion is ‘defined by example’, that is, by providing a hierarchy of descriptions based on the levels for an example UI.

Test-four: Does the context map match the reference map? *(Borchers 2001, Salingaros 2000)*

The *reference* map is created using just the reference relationships defined in the patterns. The *context* map is created using just the context relationships defined in the patterns. An idealised map can be created for a collection where every *reference*-link and every *context*-link are assumed to match.

Test-five: How 'rich' are the links within and between each level of the hierarchy? *(Salingaros 2000)*

When a language map is ordered into a hierarchy some links may link patterns *within* the same level. Other links may go *between* patterns separated by more than one level.

Test-six: Can the patterns be organised using different classification categories thereby providing alternative views? *(Borchers 2001, Salingaros 2000)*

To be classed as having internal validity, a collection of patterns should pass the first four tests. These tests define the primary attributes for describing a pattern language. The last two tests define the secondary attributes of a pattern language. Together the primary and secondary attributes describe the maturity of the language. The tests can be used to define the maturity status of a collection of patterns and will be referred to as Test-one to Test-six.

To evaluate these six tests they have been used to score several existing UI pattern collections. The following section reports on the evaluation.

3.5 Evaluating Pattern Collections

The collections of patterns that have been used to evaluate the tests in the following sections are not related to those used in the development of the tests (Todd *et al.* 2004). The UI pattern collections investigated are from those proposed by van Welie (accessed 2003) and Borchers (2001a).

The two collections created by van Welie are the graphical user interface patterns collection (GUI collection) and the web interaction design patterns collection (WEB collection) both versions accessed in 2003 (http://www.welie.com/patterns). These collections are works in progress and have undergone a series of changes over time. Not only have new patterns been added to the collection, but also the format used to describe the patterns has been modified a number of times. The versions used in these case studies were those accessed in 2003.

The final collection evaluated was the human computer interface pattern collection (HCI collection) created by Borchers (2001a). He classified his collection as a pattern language but van Welie referred to his as pattern collections. These collections have been selected because they contain a manageable number of individual patterns and they are representative of the types of UI pattern collections available at the time.

3.5.1 GUI Pattern Collection

The GUI collection contained twenty-seven patterns when analysed. The patterns are organised into six groups: Modes, Navigation, Guidance & Feedback, Presentation, Physical Interaction and Selection. The context section of each pattern defines the functional situation in which the pattern can be used without referencing higher-level patterns. Both context and reference patterns are identified in the "Related patterns" section.

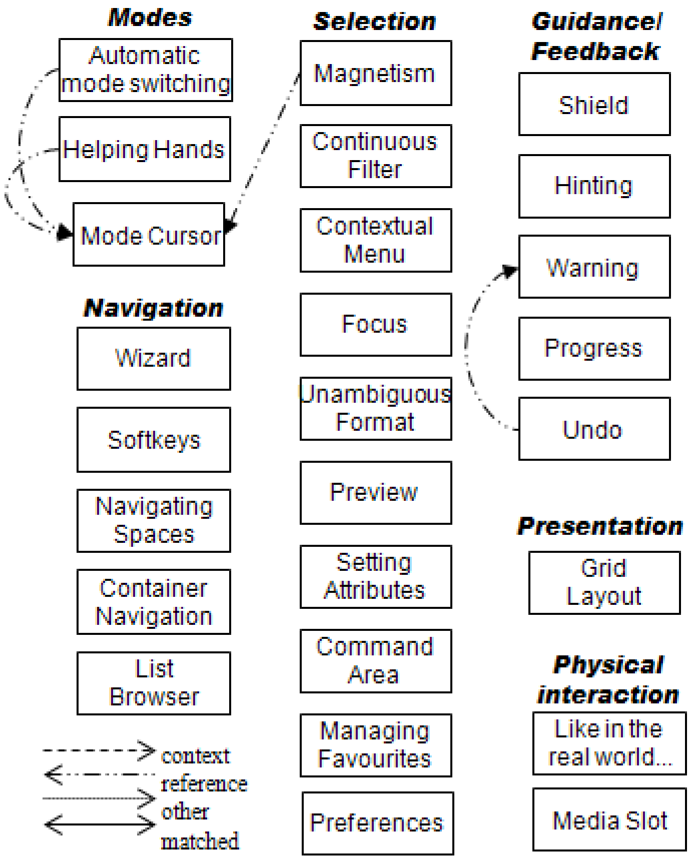


Figure 3.5 – Language Map for GUI collection.
(Website accessed – August 2003)

A language map for the GUI patterns has been created from the links identified in the 'Related patterns' sections (Figure 3.5). There are few linkages between the individual patterns; therefore the collection does not pass Test-one. This analysis indicates that the GUI collection is, as the author indicates, rather immature.

3.5.2 WEB Pattern Collection

In the WEB collection the *reference* links are found in the section labelled 'Use when' and the *context* links are found in the 'Related patterns' section. When analysed there were fifty-six patterns in the collection plus fifteen pre-patterns.

The language map is shown in Figure 3.6. The defined patterns have been organised into seven groups based on functionality; Search, Site Types, Basic Interactions, Page Elements, User Experiences, Navigation and E-commerce. There is also one reference to a pattern in the GUI collection. Pre-patterns are represented by dashed rectangles with names in italics. Nearly half of the fifty-six defined patterns are not linked into the language map. Therefore based on this pattern map the collection fails Test-one.

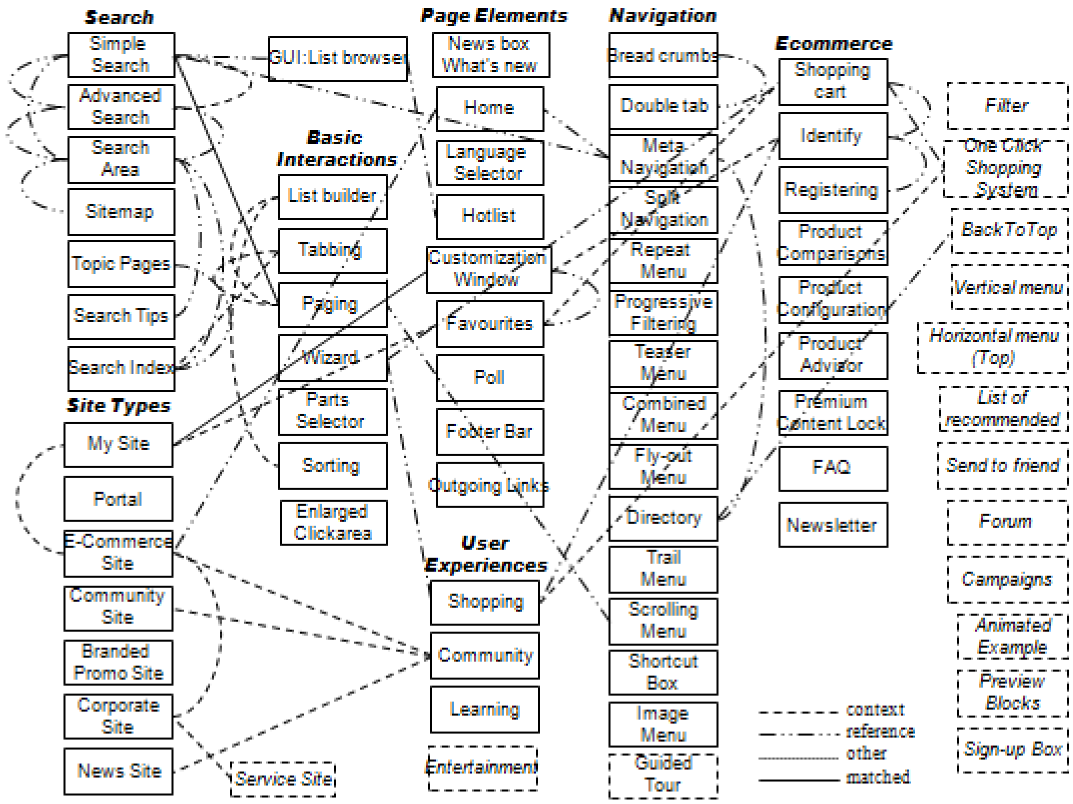


Figure 3.6 - Language Map for WEB showing just prelude and postscript links
(Website accessed - April 2003)

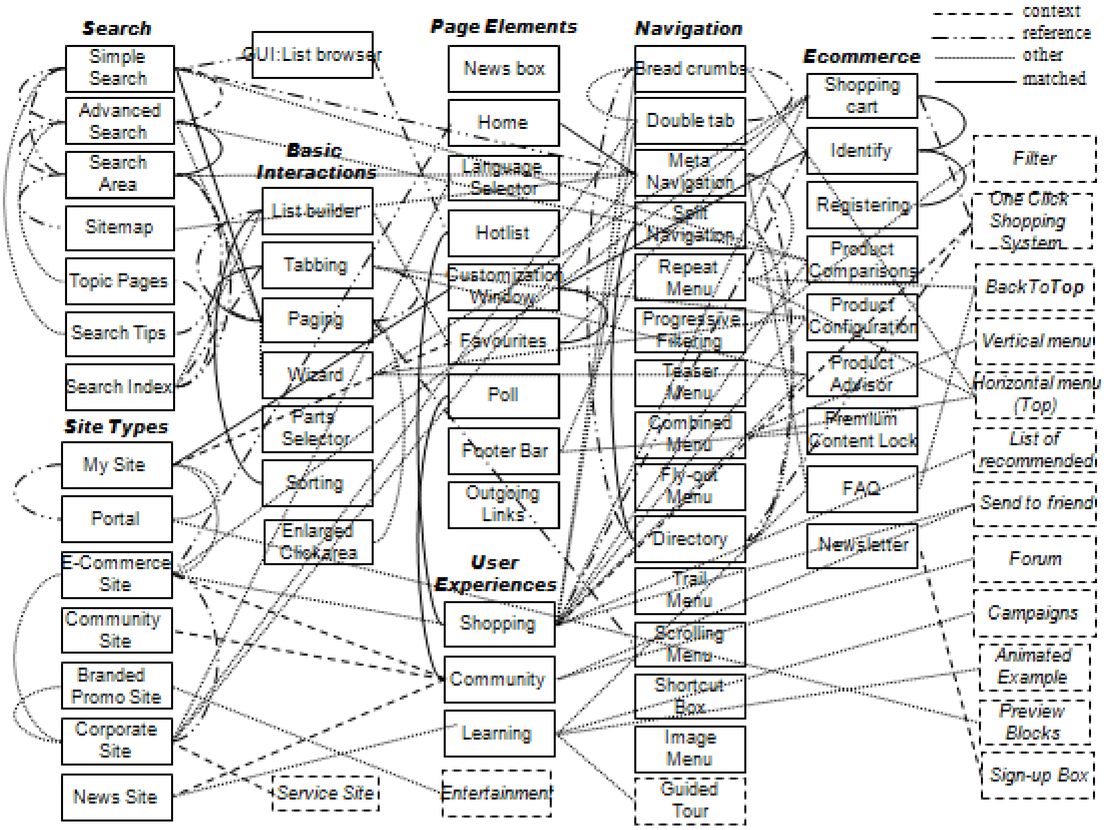


Figure 3.7 – Graph of links for the WEB collection showing all links identified
(Website accessed - April 2003)

A second language map has been created for this collection to include any reference to a pattern mentioned in any section within a pattern's definition (Figure 3.7). Most of these links are identified as ‘other’ in the diagram but where it is clear the link to the pattern complemented an existing reference or context link then the link has been classified as ‘matched’.

An examination of both the graphs created for the WEB collection (Figure 3.6 & Figure 3.7) indicates that there is no one pattern that can easily be identified as a root node for a hierarchical structure. This collection appears to be a composite of many overlapping hierarchies, one for each of the seven Site Types, e.g. the E-commerce site hierarchy. This composite collection is more complex than the GUI collection and therefore it has been evaluated against each of the validation tests in turn:

Test-one: The WEB collection could pass this test if the pass condition is relaxed because only six patterns remained unlinked on the second pattern map (Figure 3.6).

Test-two: The patterns have been organised into seven groups that may identify levels within the potential hierarchies but as yet only sub-trees and sequences of patterns have been identified. Therefore the collection does not pass Test-two.

Test-three: A pass for Test-two is a prerequisite for a pass of this test so Test-three fails. It may be possible to use patterns in the seven groups to describe different levels of detail once the collection has developed sufficiently to pass Test-two.

Test-four: Matching the context and reference maps is not possible without further analysis to define the direction of all the links represented in the graph. This analysis also raises the question as to which types of link should be included in the language map.

Test-five: In the future this test may be met. Many links are between patterns within the classification groups which may denote a degree of richness developing within levels of the collection.

Test-six: This test may be passed if hierarchies starting with the ‘Site Types’ patterns are more clearly developed. As many links in this classification are within it, it is possible that interlocking hierarchies for providing different views of a specific UI can be created. For example, a hierarchy starting with the ‘Corporate Site’ pattern includes an ‘E-Commerce site’ hierarchy (Figure 3.7).

3.5.3 E-commerce Hierarchy

There are a number of hierarchies making up the WEB collection. These can be classed as UI pattern languages in their own right. There is a precedent for this as Alexander (1979) talks about the pattern language for the garden and the pattern language for a house as being components of his architectural pattern language.

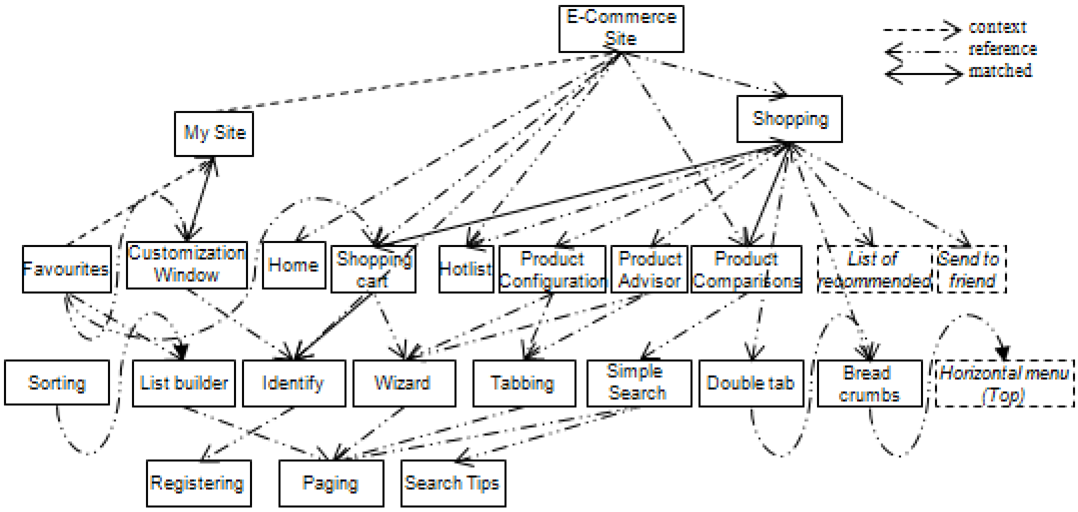


Figure 3.8 – The pattern map for the E-commerce hierarchy.

The hierarchy headed by the E-commerce pattern from the WEB collection has been selected for evaluation against the six maturity tests. This hierarchy is reasonably well developed and contains twenty-five patterns, as shown in Figure 3.8. These patterns can be organised into a hierarchical structure with a single root and four recognisable levels. The patterns in each level of the e-commerce hierarchy can be classified into one of the seven groupings used for the whole WEB collection identified in Figure 3.7.

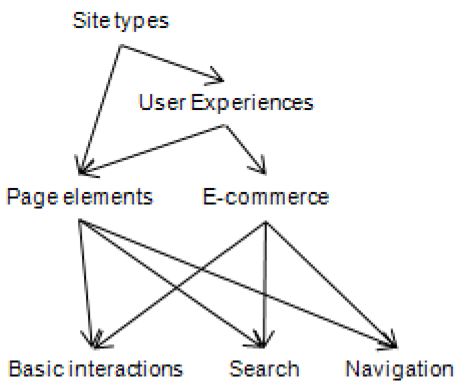


Figure 3.9 - Patterns groupings identified at each level of the E-commerce hierarchy.

Figure 3.9 shows the classes matching the levels in the e-commerce hierarchy shown in Figure 3.8. The patterns making up level three of the e-commerce hierarchy are mostly from the 'Page Elements' and 'E-commerce' groupings. Level Four of the hierarchy is composed of patterns mostly from the 'Basic Interactions', 'Search' and 'Navigation' groupings. So are the patterns in level Five, for example the 'Identity' pattern references the 'Registration' pattern both of which come from the Navigation grouping. These chains of patterns could have been modelled across the level like the patterns 'Double tab' and 'Bread crumbs' from the Navigation class. Clearly there is a great deal of flexibility in how a hierarchy could be developed when there are multiple links between patterns and links between patterns within a grouping. The e-commerce hierarchy is sufficiently complex to be evaluated against the six internal maturity tests.

Test-one: The E-commerce hierarchy passes Test-one as all members are linked.

Test-two: This test is passes because it can clearly be organised into a hierarchy.

Test-three: This test could also be passed. The patterns in each level can be used to describe a web site at different scales or levels of granularity. Figure 3.8 provides an overview of this hierarchy. The root node defines the application area then the two patterns at level one identify the main areas of the web site. The third level describes 'smaller' UI components. An example e-commerce web-site UI would need to be developed to guarantee an unambiguous pass.

Test-four: The E-commerce collection of patterns does not currently pass this test. The pattern map shows that only three of the links match. Two of the links are context only links with the majority being reference links.

Test-five: This test maybe passed because the E-commerce Site collection has a reasonably rich set of connections within each level and between levels without excessive redundancy. There is one apparently redundant link between the 'E-commerce Site' pattern and the 'Product Comparison' pattern. This pattern is also referenced from the 'Shopping' pattern, which seems to be a better representation. The two links into the 'Identify' pattern from the 'E-commerce Site' pattern and the 'Shopping cart' pattern can be justified because scenarios can easily be created where a user of a web site might like to identify themselves on first entering a site, but would not be obliged to do so until they want to use the virtual shopping cart. Creating the hierarchy is subjective, but between twenty-one percent and twenty-six percent of the links are either within or between levels depending on which links are counted.

Test-six: The E-commerce hierarchy may pass this test but determining what constitutes an alternative view is subjective. The two patterns in the first level could provide ways of describing parts of this collection from different points of view.

The hierarchy of nine patterns linked to the 'My Site' pattern can be used for viewing the UI as one that could be personalised. The seventeen patterns in the 'Shopping' hierarchy describe a shopping experience view.

3.5.4 HCI Pattern Collection

Borchers (2001) HCI collection is published with a map of the language (ibid, Figure 3, page 375). The pattern descriptions of these seventeen patterns are similar to the Alexandrian format (Alexander *et al.* 1977). Each of the patterns is identified by number but although the root is identified by 'H1' the higher numbers do not identify patterns only in the lowest level as in Alexandrian pattern language (ibid). For example, the pattern 'Flat and Narrow Tree' which occurs on level three below the root has the identifier 'H7'.

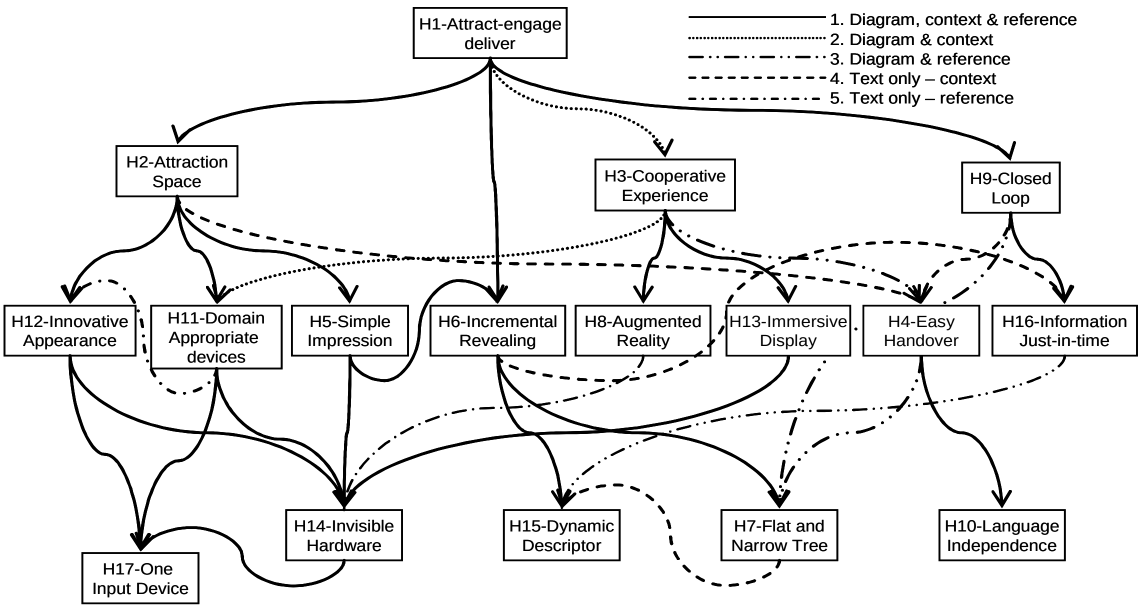


Figure 3.10 - Idealised HCI collection language map combining all sources of link information.

The published map (Borchers 2001) has been compared with the definitions of the individual patterns by matching the context and reference links found in the pattern definitions with the arcs on the pattern map. The types of line representing the links on the diagram in Figure 3.10 are the result of this analysis. Five states have been identified:

1. The link appears on the diagram and the context and reference relationships in both patterns match.
2. The link appears on the diagram and is defined as a context link in the appropriate pattern.

3. The link appears on the diagram and is defined as a reference link in the appropriate pattern.
4. There is no matching link on the diagram but a relationship has been defined as a context link in the pattern.
5. There is no matching link on the diagram but a relationship has been defined as a reference link in the pattern.

There are no links represented on Borchers diagram that are not also defined in one or more of the related patterns. However there are some relationships defined in patterns (items four and five above) that are not shown on the diagram. Although the HCI collection is relatively small its structure justifies testing for maturity.

Test-one: This test is passed because all seventeen patterns are linked into the pattern map.

Test-two: This collection of patterns is presented as a pattern map that is structured as a hierarchy so it passes Test-two.

Test-three: This collection has been successfully used to describe a user interface (Borchers 2001) so it passes this test. Even so, when reading across Level 1 of the language map, there appears to be a problem with the positioning of pattern H6 ‘Incremental Revealing’. Should it be part of Level 1, or in Level 2 as shown in the pattern map (Figure 3.10). A reading of the three patterns (H2, H3 and H7) that make up Level 1 indicates that a high-level description of the interface is possible without including pattern H6.

Reading across this level indicates that the designer first, considers specifying how the system will look when first encountered so that it will attract the attention of potential users. Secondly, considers how to involve more than one user in a co-operative activity. Thirdly, how a user can pass control to another user and leave the exhibit has to be determined. Adding the description for H6 certainly makes the intention of the design clearer at this scale but the pattern appears to fit equally well in Level 2. H6 seems to fit the property of self-similar scaling described by Salingaros (2000).

Test-four: If applied rigorously, the HCI collection does not currently pass this test.

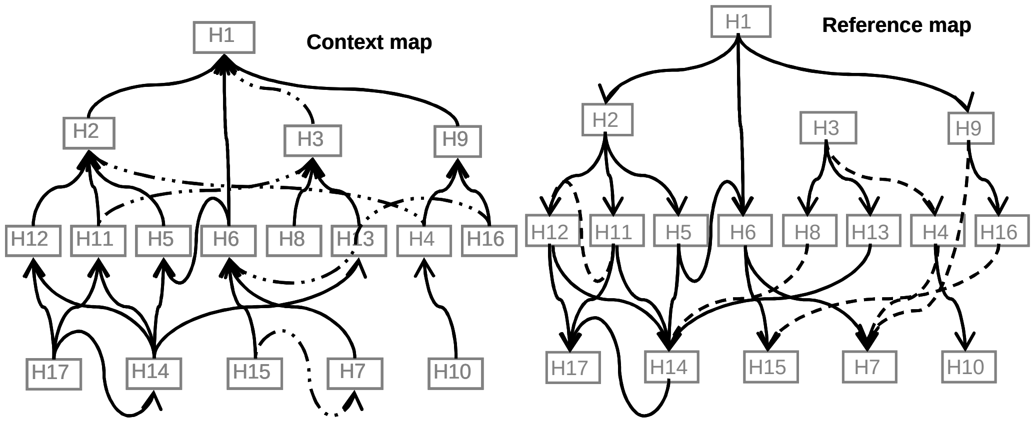


Figure 3.11 - The Context map and Reference map for Borchers HCI patterns.

All relevant links identified in both the patterns and the pattern map, have been used to create the context and the reference maps shown in Figure 3.11. A comparison of these two maps shows that they do not match therefore this collection fails

Test-five: The HCI collection has the potential to pass Test-five. Most links shown on the pattern map are between nodes on different levels in the hierarchy but twenty-three percent of the links are within two of the levels indicating that the language is developing a degree of richness.

Test-six: This test is failed. There is only one root and there does not appear to be any alternative way to view the patterns in this language. Another limiting factor of this collection is that it is probably too small to be classified as mature.

3.6 Summary

Students reportedly have difficulty with understanding and using the structure of a pattern language (Dearden *et al.* 2002a, b, Wesson & Cowley 2003, Kotzé *et al.* 2006, Koukouletsos 2008). To help students overcome this learning problem an educator needs to understand pattern language structure. To that end this chapter investigated the internal validity of pattern languages. Three types of validation have been identified as requiring consideration when assessing the characteristics of a pattern language: the validity of the individual patterns, the internal validation of the pattern language and the external validation of the pattern language. Pattern language internal validity relates to the structure of a pattern language.

A set of six tests have been proposed for evaluating internal validity, richness and views of a pattern language's structure. These questions have been used to evaluate four existing collections of patterns none of which passed all tests. A responses to the first four questions can be a simple '*yes*' or '*no*' for determining internal validity. But, when applied to evaluating a pattern language one hundred percent compliance seems to be unrealistic. UI pattern languages should be developing over time with use and especially as technology changes. Therefore links should be evolving even in a well developed pattern language.

Determining a pass for the other two tests is less clear and a set of criteria describing the state of richness or views would be more useful. Because many patterns are written without making clear distinctions between the different types of relationship linking them to other patterns, the sets of criteria should include a response labelled '*possibly*' for well connected pattern languages where relationship types have mostly not been analysed. A carefully analysis of such relationships could change a pattern language's internal validity state.

Chapter 4: UI Pattern Language Maturity Model

The development of a set of six tests for evaluating the internal validity, richness and views of a pattern language (PL) was described in the preceding chapter. Although the application of the tests to evaluate four existing patterns languages indicated that the process would help educators gain knowledge about the structure of a pattern language the pass criteria were not particularly helpful for comparing pattern languages. In this chapter the tests are revisited and sets of criteria defined to better characterise the structure of a pattern language.

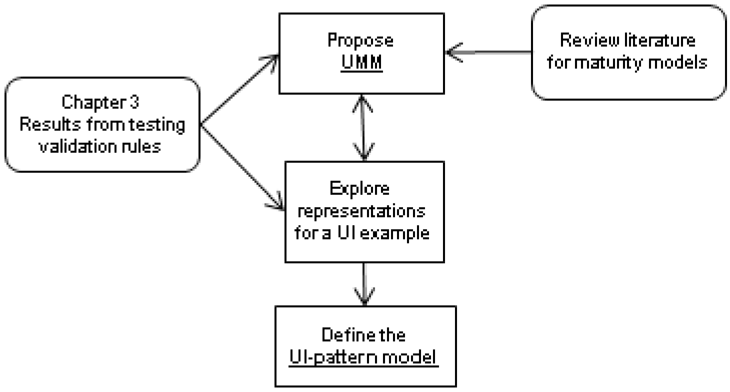


Figure 4.1 - Process followed for this investigation

Figure 4.1 shows the dependency of this chapter on the tests developed in Chapter Three. The tests are evaluated and a set of pass criteria for the revised set of seven tests is presented. A technique for combining patterns into a hierarchal structure that describes an example UI has been defined and named a UI-pattern model. A discussion follows proposing that building these models can be used as a generative process for advancing a UI pattern language's structure. A short review of maturity models follows which leads to the description of the proposed UI maturity model, based on the revised set of tests. This maturity model is evaluated by using it to rate each of the pattern languages introduced in Chapter Three.

4.1 UI Pattern Language Structure

Assuming that the four UI pattern collections introduced in Chapter Three are a representative sample then applying strict criteria for passing each of the first four tests of a pattern language's structure would result in none of them being classed as internally valid. This observation appears to support Salingaros (2000) who comments that:

> *"The most elegant complex systems are nearly (but not perfectly) ordered." (ibid, p159)*

It makes sense that when developing a pattern language it will not be perfect but neither will a well developed pattern language. Language structure should not be static, responding over time through usage, and thus maturing. This is a generative process (Alexander 1979). The proposed tests define an idealised UI pattern language but a hundred percent pass is unrealistic. To be useful as a measure of an existing pattern language's maturity they need to be relaxed with less stringent passing criteria defined. Where possible terms like 'nearly', 'not perfect' or 'most' need to be quantified. These changes should enable the structural maturity of UI pattern languages to be compared.

From applying the tests it is clear that the first four are directly related to the fundamental structure of the pattern map. They measure the pattern languages' internal validity by evaluating the primary structure of a pattern language.

The last two tests relate to links and structures that are structural refinements. Test-five evaluates links within a sub-group of patterns and identifies alternative relationships. Test-six is about evaluating the structure can be interpreted or provide meaning. These two tests relate to secondary structures within a pattern language. Together, the tests of the primary and secondary structures of a pattern language characterise the maturity of a pattern language. In the following discussion the results of evaluating the four pattern collections are combined to determine practical criteria against which a pattern language's structure can be assessed.

4.1.1 Primary Pattern Language Structures

The first four tests assess internal validation by characterising the primary structures of a pattern language, and are discussed below:

Test-one: Do the reference and context links between the patterns form a map?

The studies reported above resulted in the HCI collection and the E-commerce collection both passing Test-one. In the WEB collection thirty-seven patterns and pre-patterns are not linked into the language map when it is created using clearly defined context and reference links (49% connectivity). Creating a map using all links mentioned in the pattern definitions reduces the number of unlinked patterns down to seven (90% connectivity).

The percentage of unmatched links that can be tolerated for a pass needs to be determined. A connectivity rate of forty-nine percent seems insufficient for a pass but a rate of ninety percent could be taken as sufficient,. By accepting this rate van Welie's WEB collection could pass Test-one.

Which types of link should be considered when developing the language map? Including all links to other patterns within the body of each pattern's description raises a second issue to address – should all relationships to other patterns in the collection be included in the language map?

Eight main groups of relationship type were identified in the literature review (Table 2.4). Schümmer and Lukosch (2007) consider all link types should be shown in the pattern map but they were interested in the richness of a language not its internal validity. In PLML (Fincher 2003a) the workshop participants of UI experts only agreed to three types of link: *contained*, with its inverse *is-contained-by* and *is-a*. These are the basic relationships required for structuring a pattern language and provide an acceptable limitation for this test.

For internal validity, the ***reference and context links between the patterns form a map*** *of relationships as defined in PLML and at least ninety percent of the patterns should be linked into the network.*

Test-two: Can the pattern map be ordered into a hierarchy of levels?

There is no discernible primary hierarchy in van Welie's GUI collection. The WEB collection is organised into groupings which could be levels and there appears to be a set of interlocking hierarchies based on the 'Site Types' groupings' patterns. One of these hierarchies, the E-commerce hierarchy, has four levels below the root and these equated to the organisational groupings as shown in (Figure 3.9). The HCI collection is a single hierarchy and has three levels below the root. How many levels below the root node should the hierarchy extend for a collection of patterns to have internal validity? The HCI, the WEB and E-commerce collections could all pass this test if the minimum number of levels specified is three.

There are composite pattern languages where there is no primary root and a set of interlocking hierarchies based on one or more pattern categories (e.g. the WEB collection). This type of organisation raises questions about how many hierarchies are required to pass this test. Should one hierarchy be nominated as the primary hierarchy? Should a generalised pattern be created to become the parent for the set of roots when multiple hierarchies can be identified?

In a mature UI pattern language it would not be necessary and may not be desirable to have all patterns linked into a single hierarchy because the richness of the language may be partly represented by the inter-relations between alternative hierarchies. There does not appear to be any valid reason for having one dominant hierarchy. In some pattern languages one hierarchy may be dominant but this may not be necessary for internal validation, for example in the WEB collection, none of the patterns in the ‘Site types’ group is more important than another, they are simply different.

A pattern language should continue to develop with use over time (Alexander 1979). At any moment, the hierarchies in a composite pattern language will probably be at different stages of development, for example the placement of a particular pattern may not initially be clear and can change over time. In the absence of any other evidence an arbitrary value of ninety percent connectivity has been chosen. Further evidence may provide a value better supported by either research or practical experience.

For internal validation the language map can be organised so that at least ninety percent of the patterns in it can be placed into a recognised level in one or more hierarchies and at least three levels can be identified below the root level.

Test-three: Can the levels be used to describe a user interface at different degrees of granularity?

The hierarchy that orders the Alexandrian pattern language (Alexander *et al.* 1977) is based on a well-understood scale of granularity, from a cityscape down to architectural features of individual doors and windows. A comparable 'natural' organisation for user interfaces has yet to be identified. This topic was discussed by participants at the Usability Pattern Language Workshop at CHI2002 (McInerney 2002). Mullet (2002) highlights the problem of defining degrees of granularity in his discussion of pattern language organisation saying:

> *"The murky ground between application-level and widget-level patterns is the terrain that must be addressed by an organisation scheme." (ibid, p4)*

He also considers the "*level of design activity that the individual patterns described*" as potential levels of granularity. These identify conceptual-level patterns that describe content and organisation, patterns that describe the behaviour or feel of the interface and patterns describing the appearance or look of the interface. These levels of granularity describe UI design with different degrees of abstraction (Seffah & Forbrig 2002, Märtin & Roski 2007).

Different UI pattern collections cover different areas of UI design. Schümmer and Lukosch’s (2007) patterns are organised into three levels of granularity: Community Support, Group support and Base Technology. Each of these levels is further sub-divided into groups of patterns. The patterns created by van Duyne *et al.* (2003) are organised into twelve different groups ranging from the very general descriptions of the ‘Site Genres’ patterns down to technical patterns in the ‘Speeding up your Site’ group. Borchers’ HCI collection (2001a) of seventeen patterns has a narrow scope. Each of the three levels below the root of the HCI collection provides a description of a multi-user interactive kiosk interface. The E-commerce sub-system (van Welie, accessed 2003) ranges across more levels of abstraction than the HCI collection. The root describes the application area (Site Type) down to abstract widgets such as "Bread crumbs".

From the small sample of UI pattern languages discussed above, hierarchies could range from three levels to a maximum of twelve levels of granularity. The UI patterns in these address issues ranging from highly conceptual problems to highly technical. To pass Test-three, not only does the classification system used to organise a pattern collection have to clearly identify different levels, an example model of an actual UI should be created. This example should demonstrate that descriptions of the UI could be generated at different levels of abstraction.

Both the E-commerce collection and the HCI collection can be described at different levels of abstraction. Borchers (2001a) provides an example that uses every pattern in the HCI collection, so it passes this test. For the E-commerce collection to meet this test, an example model of a UI would need to be created.

For internal validation, the patterns at each level should describe a different level of granularity. This should be demonstrated with an example.

Test-four: Does the context map match the reference map?

The idealised pattern language map includes all of the links defined in both the context map and the reference map. Therefore, the idealised pattern map has all context and reference links matched.

When developing a UI pattern language, checking that the context and reference maps match is a mechanical process. Even in a situation where the author of the pattern language has identified this criterion as important (e.g. Borchers 2001a) it appears to be difficult to attain in practice (Figure 3.11). Why are there discrepancies between the context and reference maps, even where the author of the patterns has clearly stated that consistency between these two maps is important? One clue may exist in the numbering of Borchers' pattern language which includes patterns with higher numbered identifiers than their positions in the hierarchy would suggest appropriate (Figure 3.10). For example, the reader would expect that the pattern 'H9 Closed loop' in level one would be 'H4 Closed Loop'. Such discrepancies indicate that development of this pattern language went through a process which Salingaros (2000) called 'repair and replacement'. This is an evolutionary process where a language changes and matures over time. Keeping all links between patterns consistent as more patterns are added to a pattern language and other patterns are modified probably needs a computer-based pattern management system. A system such as MUIP (Deng *et al.* 2006) enables changes to be more easily be tracked.

Both Salingaros (2000) and Borchers (2001a) acknowledge that ideally all links should match but that it is not a necessary condition. This view raises the question– what proportion of the links should match for a collection to have internal validity? There are thirty-two links on the idealised language map for the HCI collection but the Context map and the Reference map identify two different sets of 26 links. That is an 81% match to the idealised map. Because this collection had the best matched connectivity of those analysed, up to a twenty percent miss-match in matched connectivity has been chosen as the pass criteria.

For internal validity the difference between the context map and the idealised map is less than twenty percent and similarly between the reference map and the idealised map.

A set of criteria is required to quantify terms like 'nearly', 'not perfect' or 'most' used in descriptions for pattern languages (e.g. Salingaros 2000)., for example at least ninety percent for Test-one or less than twenty percent for Test-four define a pass.

4.1.2 Secondary Pattern Language Structures

Another three tests assess a pattern language's secondary structures. In the following discussion these tests are considered along with possible pass criteria. They evaluate the secondary structures found in a pattern language by using all the identified relationships between the patterns and any alternative views.

According to Salingaros (2000) a pattern language's maturity is partly determined by how rich the network of links between the patterns in the language has become. He mentions that both inter- and intra- level links contribute to richness. He also discusses different types of link when discussing pattern language maturity. For determining internal validity, the links for creating a pattern map have been restricted to the three basic types. This pattern map can be referred to as a *validation pattern map* defining the primary structure of the pattern language. When considering the maturity of a pattern collection all other relationship types as defined in Table 2.4 should be considered as they indicate the richness of a pattern language.

It should be possible to generate alternative views (Salingaros 2000) when a mature pattern language is used to describe an artefact, for example a UI. When using the Alexandrian patterns in architectural design, patterns may be selected that focus on social issues, in contrast to those resolving spatial problems of building design. UI patterns exist that focus on different dimensions of UI design, for example addressing social, behavioural or temporal issues, indicating the possibility of creating alternative views.

The proposed pass criteria associated with each of the tests are arbitrary but are based on the experience gained from the analyses of existing pattern languages reported above (Section 3.5). However over time the expectation is that all the scales will be refined as further pattern languages are analysed.

Test-five: How 'rich' are the links within and between each level of the hierarchy?

In the first version of this test, richness is characterised by the intra-level links that occur between patterns *within* a level and inter-level links that *traverse* more than one level. The GUI collection has only four links between patterns, two within the ‘Modes’ pattern group, one in the ‘Guidance/feedback’ group. The other is between two patterns, one in the ‘Modes’ group and the other in the ‘Selection’ group. As a hierarchy cannot be created with these patterns even though three of the four links are within groups, this collection cannot be rated as having a rich set of connections. To consider a collection of patterns as mature its connectivity should be reasonably well developed.

All the links used to create the pattern map for the HCI collection could be classified as basic types. Over twenty percent of the links are intra-level links and occur in two of the three levels below the root. Two inter-level links have been identified between levels. Only two patterns could be connected by one of the other types of relationship, for example, Pattern ‘H8-Augmented Reality’ has an ‘alternative’ relationship with ‘H7-Flat and Narrow Tree’ where the pattern chosen will be based on how the designer wishes to structure the content to be displayed.

There are five intra-level links in the E-commerce hierarchy, thirteen percent of all identified links. Seven inter-level links have also been identified, comprising eighteen percent of all links. A number of alternative relationships between patterns have also been identified, for example the '*variant'* relationship between the three list-related patterns 'List of recommended', 'Hotlist' and 'Favourites'.

It is clear from the preceding discussion that 'richness' is evaluating two different attributes: 'complexity' and 'richness'. Complexity is a measure of the amount of connectivity within and across levels in the hierarchy. Richness now becomes a measure quantifying the number of non-basic relationships identified between patterns.

A six point scale has been created for evaluating the '*complexity'* of the connectivity of a pattern language. The details are in Appendix A1. When creating this scale, one issue addressed is that a complex structure might indicate a mature language structure. Alternatively, it could also indicate a poorly understood structure, as demonstrated by the Common Ground example (Figure 3.3). Therefore the complexity scale requires that the pattern map includes identifiable hierarchies. For example, to be classified as '*Developed'*, a pattern language's complexity is defined as '*one or more hierarchies can be identified but less than thirty-five percent of the links are intra-level links and inter-level links*'.

To evaluate a pattern language's '*richness'* a five-point scale has been developed as described in Appendix A1. The richness units of '*Developing'* and above require the connectivity of the pattern map to be at least ninety percent. Each unit identifies the percentage of non-basis relationships required, for example to be classified as '*Developed'*, a pattern language's richness is defined as '*at least ten percent of all relationships are of non-basic types*'

The wording for the two new tests is:

New Test-five: **How 'complex' are the links *within* and *across* levels of the hierarchy?**

Scale Units = (None, Minimal, Possibly, Developing, Developed, Complex)

New Test-six: **How 'rich' are the relationships linking the patterns?**

Scale Units = (None, Possibly, Developing, Developed, Rich)

Test-seven[3]: Can the patterns be organised using different classification categories thereby providing alternative views?

Salingaros (2000) says that a pattern language "*has multiple tops and horizontal connections*". This implies that an alternative view will start with an alternative root. Is this a valid proposition? Associated questions to consider are: What percentage of the patterns is required to define an alternative view? Should alternative views be mutually exclusive or can hierarchies overlap?

A number of authors have proposed alternative schemes for categorising the patterns in their pattern collections (e.g. van Welie & Traetteberg 2000, Tidwell 1999). These take the form of lists of patterns in each category. Are these levels in the language or different viewpoints? Possibly there is a mixture of both. For example, when using van Duyne *et al.*'s WEB patterns (2003) the 'Writing and Managing Content' patterns in group D are more likely to form a level within a hierarchy. On the other hand it may be possible to create a hierarchy of patterns from the 'Creating a Navigation Framework' patterns in group B to form a 'navigation' view. The patterns in this view will overlap at least one other hierarchy and the view will 'start' below the root pattern defining the site genre. Patterns from both the Site Types grouping and the User Experience grouping could be used to identify different views as demonstrated with the E-commerce hierarchy. This hierarchy could be viewed, from the viewpoint of, either a user personalising it, or the shopping experience. For the purposes of this test a view is defined as:

- Providing a way to describe a UI or part of it from a particular focus such as navigation or usability,
- Being a subset of the patterns that describe a UI and organised into a hierarchy,
- Overlapping with one or more alternative views,
- Requiring a root pattern but this root may be within the context of other patterns not included in the view.

The number of views that can be found in the pattern language will indicate how mature the language is. The theoretical range would be from none to some number less than the number of patterns in the pattern language. But the pattern language does require a reasonable degree of connectivity.

[3] This test was labeled 'Test-six' in the earlier versions of the tests.

The scale for indicating the number of alternative views is: (*None, Possibly, Developing, Developed, Views*)

4.1.3 Summary

The rules defined by the seven modified described tests above, provide reasonable grounds to argue that every collection of patterns should be referred to as pattern languages rather than a collection, set or catalogue (Appleton 1997). Pattern languages can be characterised by identifying distinguishing attributes of their primary structures (Validation mapping, Hierarchical structure, Descriptive Levels, Matching Context & Reference maps) and secondary structures (Complexity, Richness & Views).

4.2 Patterns Representing a UI Example

Test-three focuses on levels of abstraction that should be apparent in the organisation of a pattern map. To pass this test, an example UI needs to be developed to demonstrate that descriptions of a UI at different levels of granularity can be created.

The methods for selecting patterns that describe a UI suggest forming the selected patterns into a list (Tidwell 1998, van Duyne *et al.* 2003). Applying Alexander *et al.*'s method (1977) also results in a list of pattern names which he calls a sequence, identifying them as the key to a good design. Correctly formed sequences or pattern lists are the key to a good design. Essentially this involves moving from a higher to a lower level of detail, although sometimes the sequence:

> *"follows a line, dips down, dips up again" (ibid, pxviii).*

When using architectural patterns to teach groups of students about aspects of urban design, Davis (1983) identified the sequence of selected patterns as crucial to the design process. He also recognised the importance of identifying 'global' patterns in this sequence, effectively acknowledging the hierarchical structure of this list of patterns, by commenting

> *"The sequence of patterns was also considered, since the actual design would also take place in group discussion, requiring understanding of which patterns were more global, and therefore needing to be designed first." (ibid, p15)*

Walldius (2003) considers the list of selected patterns tells a 'story' describing the UI. He also recognises that it is not clear when using just the list, how the patterns fit into a hierarchy or which relationships between patterns are relevant to a specific problem. In his 'story' example, relationships and hierarchy still have to be located within the narrative. To demonstrate a pass for Test-three some form of 'story' needs to refer to different levels of description. This requirement indicates a need for a form of presentation that will:

1. Record the patterns selected to describe a UI.
2. Identify a hierarchy representing different degrees of granularity.
3. Identify all required relationships between patterns.

The following sections report on the development of a diagram that describes an example UI using the selected patterns better than a list of selected patterns.

4.2.1 UI-pattern Model for the Outlook Today view

The list of patterns from the Common Ground pattern language (Tidwell 1998) representing the Outlook Today example is found in the PATTERNS column of Table 3.4 (col. 2). The list is grouped, based on Tidwell's questions for guiding selection of patterns. This provides no sense of how the patterns relate to each other. The pattern map for the same subset shows the relationships linking the patterns (Figure A2.1) using all links identified between the participant patterns plus those that could be inferred from the text. As a description it is not particular helpful. This hierarchy of patterns can be reorganised to create a structure that better describes the Outlook Today view. This new structure has been named a 'UI-pattern model' and a full description of how it was constructed can be found in Appendix A2.

A UI pattern model is a graph containing one or more interlocking hierarchical structures linking the selected patterns together, where the backbone of the primary hierarchy depicts the structural elements of the UI. The UI pattern model forms a meta-language for a UI (Erickson 2000) describing the UI at different levels of abstraction. It is a UI modelling tool similar to that envisaged by Seffah and Forbrig (2002) who said:

> *"Developers can use one high level language to implement an abstract and device independent user interface model." (ibid, p125)*

To convert a pattern map into a UI-pattern model the designer needs to identify the candidate root patterns for at least one hierarchy and identify which links to include and which to discard. In addition, missing links need to be identified and added, as do missing patterns.

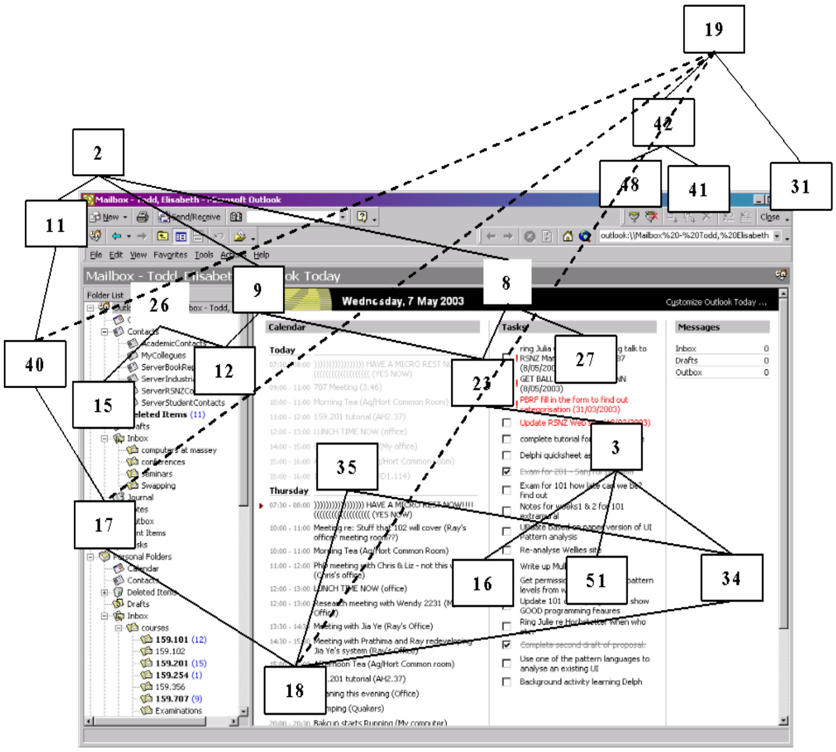


Figure 4.2 – Graph of patterns overlaying example UI with dotted lines representing relationships interlocking the two hierarchies

Figure 4.2 shows the start of building a UI-pattern model for the Outlook Today view. This view displays relatively complex information so Pattern 2 has been selected as the most likely root. The other patterns have then been overlaid on an image of the UI by placing the pattern identifying numbers (Ids) near the parts of the UI that they represent. This process helps to identify relevant relationships so that apparently unnecessary links can be removed. It may also identify missing patterns if a suitable pattern has not been found to describe some part of the UI.

Most relationships on the partial pattern map, as shown in Figure 4.2, have a spatial component but some are temporal, for example there is a chain of links from pattern 40 to pattern 17 'Pointer shows affordance' and then onto pattern 18 'Short description'. The user would perceive the order to be first the pointer showing affordance when coming into the space of a tool and second a short description appearing. A small group of patterns linked to pattern 19 'Sovereign posture' form a second hierarchy. According to Tidwell (1998) this type of pattern is concerned with a user's attention and defines a mixture of behavioural and social aspects of a user interacting with the software.

Three types of missing link can be identified from the restructuring of the partial pattern map into a UI-pattern model. They are:

1. Missing context links; found by identifying patterns with no context relationship linking them into the hierarchy. A new relationship needs to be created linking the pattern to another pattern that best describes the surrounding interaction space, for example pattern 35 'Choice from a large set' will be linked to pattern 12 'Hierarchical Set'.

2. Missing reference links. Based on observed behaviour and placement of widgets identify any missing reference links where the set of patterns on a level doesn't provide a complete description. For example, the pattern 31 'Convenient environmental actions' needs to be linked back to pattern 2 'High-density information display' to represent actions that can be carried out by the main window such as resize, close and minimise.

3. Amended links. Both patterns 34 and 35 are linked to pattern 18 'Short Description', but the actual behaviour observed in the UI 'Outlook Today view' shows the lists representing the Calendar, Tasks and Messages should be linked to pattern 17 'Pointer shows affordance'.

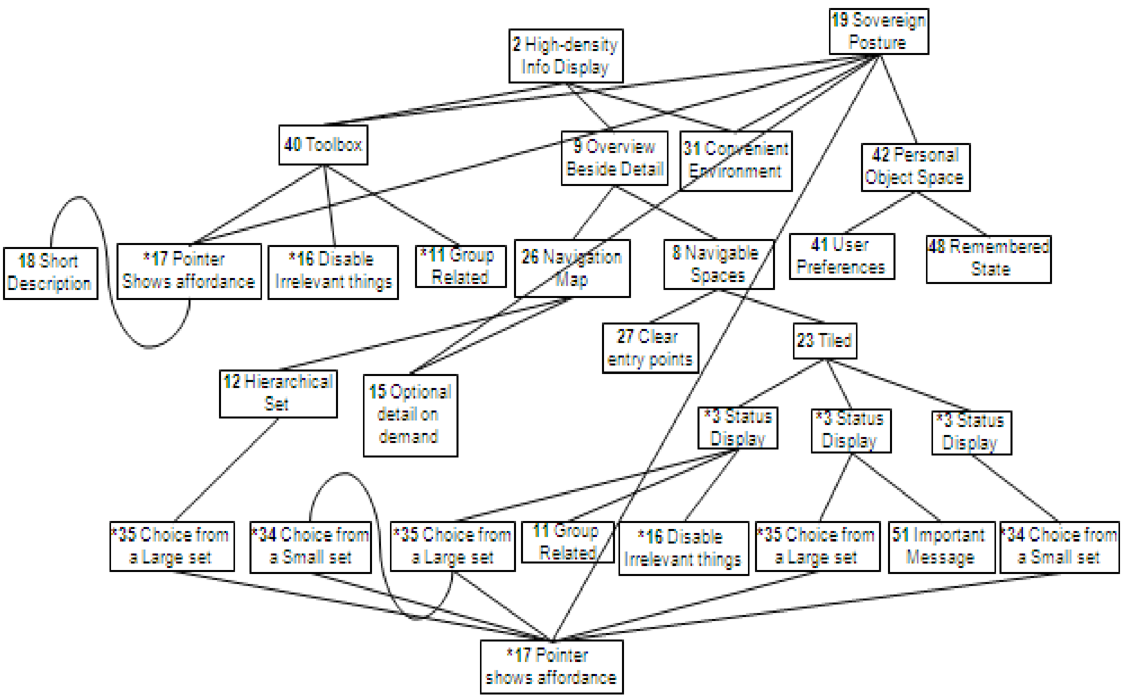


Figure 4.3 - UI-pattern model for the Outlook Today UI example

Patterns have been duplicated where they are required to describe a distinct feature of the UI. The completed UI-pattern model is shown in Figure 4.3. Duplicates are prefaced with asterisks, added to improve readability, for example pattern 3 is repeated three times representing the spaces required for the Calendar information, Tasks and Messages respectively.

The descriptions generated by the levels and within sub-trees based on the UI-pattern model have a strong spatial component. For example, starting from level one the screen from Outlook (Figure 4.2) can be described as having three main interaction spaces. First pattern 40 ‘Toolbox’ represents the space where primarily toolbars are found. The second space represents the main structure of the information. It is represented by pattern 9 ‘Overview beside detail’ identifying the two main panels as shown in the example (Figure 4.2). The third space represents convenient actions such as ‘Close’, ‘Resize’ and ‘Minimise’.

Although not covered by this example, it should be possible to use the UI-pattern modelling process to identify missing patterns as well as missing relationships. If pattern 31 ‘Convenient Environment’ did not exist then there would be no patterns for representing the required actions. The changes required to the UI-pattern model to provide a complete description of the UI can subsequently be used to update the pattern language.

4.2.2 UI-Pattern Modelling as a Generative Process

A number of authors (Chung *et al.* 2004, Lewis *et al.* 2004, Märtin & Roski 2007, Saponas *et al.* 2006, Wania 2008) described processes for identifying the structure of a collection of patterns. They all described a process for re-structuring a pattern language by examining the patterns and identifying relationships

Wania (2008) describes how the aIRPlane pattern language was created including investigating its structure. Like other researchers (Chung *et al.* 2004, Lewis *et al.* 20047, Saponas *et al.* 2006) she used card sorting (Cooke 1994). The results of having seventeen participants use card sorting to organise the patterns, were analysed using hierarchical cluster analysis, factor analysis and multi-dimensional scaling to create the basis of a network structure for the aIRPlane pattern language.

Märtin and Roski (2007) define a rule-based workflow approach for developing the structure of a pattern language. It uses problem decomposition as the main method for defining a pattern language structure that is strongly hierarchical with only a few links between sub-trees being considered appropriate. According to their rules, each level in the structure should include patterns solving a sub-problem with the same degree of granularity or detail. The root pattern describes a whole problem in general terms. They provide rules for identifying relationships that need to be modified or deleted, and how to identify where new relationships are required. Their process can also identify missing patterns. The results should be a validated pattern language structure but it may lack maturity features such as complexity. For example, links within a sub-tree crossing more than one level and links between patterns at the same level are not allowed.

The Alexandrian concept of a generative process (Alexander 1979) is one where a pattern language is refined over time by repeated use in guiding the design of new cities, commercial buildings and dwellings where inconsistencies and omissions are identified and remedied. The processes described above are not generative processes because they do not base their structuring techniques on designing or describing actual examples using the patterns from the relevant pattern language. The modification of a pattern language identified by building UI-pattern models for different user interfaces is generative.

The generative nature of a pattern language can be exhibited in a number of different ways. Dearden and Finlay (2006) identify four other ways in which a pattern language is generative:

- Individual patterns because the advice they contained can be used to improve many given UI designs,
- The development from one pattern language of many sequences of patterns representing different UIs,
- A sequences of patterns that form a meta-language with supporting theory,
- From each pattern sequence many instances of physical UI designs can be generated.

The development of the UI-pattern model described in Section 4.2.1 used a subset of Tidwell's Common Ground (1998) pattern language. She pointed out that the link structure of this pattern language (Designing Interfaces) was incomplete. An analysis of a subset of her updated pattern language (Tidwell 2006) reported by Märtin and Roski (2007) demonstrated that its structure is also incomplete with many patterns being isolated. Analysis of the pattern map they constructed shows that after seven years of development the connectivity of this pattern language is seventy-seven percent. The creation of the UI-pattern model for the Outlook Today UI example identified missing and inappropriate links. These have been corrected and those corrections could feed back into the original pattern language and over time the structure of latter versions of the pattern language should evolve and mature in the manner that Alexander envisaged.

As well, the UI-pattern model provides an enhanced graphical representation of the pattern sequence and is a meta-language from which many physical UI designs can be generated.

4.2.3 Advantages of the UI-pattern Model

The UI-pattern model overcomes the limitations of using only the pattern list. Being a graphical representation the hierarchical structure of the list can be identified. This is important when using a pattern language as part of UI design because:

> "*Whereas an individual pattern provides a solution to a specific problem, organised as a hierarchy, patterns guide the developer through the entire architectural design.*" *(Seffah & Forbrig 2002, p131)*

The advantages of using the UI-pattern model are:

1. The UI-pattern model is a visual presentation of the patterns describing a UI.
2. The patterns are organised into a hierarchy representing different degrees of granularity
3. The relationships between patterns that describe the UI being modelled are clearly identified.
4. The UI-pattern model aids in the identification of pattern definitions and link structures that could be improved and where new patterns or relationships might be required.
5. The model can guide the UI developer through the UI design process.
6. The results from developing UI-pattern models can be used to update a pattern language, providing a process by which a pattern language can mature.

Another potential advantage is that an educator can demonstrate the basic principles of using a pattern language's structure by developing a UI-pattern model. As shown with the single screenshot example, Figure 4.2, only a manageable subset of a language is required to build a reasonably complex UI-pattern model. For an educator this approach also has the advantage that students can work with examples of real UIs, rather than just abstract ideas when being introduced to a UI pattern language.

4.3 Maturity Models

A maturity model is one that provides a set of criteria by which the evolutionary development of the focus of the model can be assessed. For example, the Capability Maturity Model (CMM) provides measurement tools and techniques for assessing an organisation's IT development processes (Herbsleb *et al.* 1997). Suggate's Rank of New Zealand coals (Suggate 2000) is a model for assessing their maturity. In business the process maturity model is a tool that can be used to guide incremental improvement in the specified process. In their simplest form a maturity model identifies a set of maturity levels and a set of goals for each level that identify the factors that can be used to characterise the focus of the model. For example the set of factors or attributes, and the values they should attain at each level as defined in the basic coal rank table, determine the maturity of a coal measure (Sherwood & Phillips, updated 2009). The CMM is a well developed model, described as a reference model that can be use by an organisation to guide the incremental improvement of their IT development processes. It provides a set of factors and measures, that an organisation can use to rate its software development processes and also provides measures for comparing processes from different companies *(*Herbsleb *et al.* 1997). A maturity model is not static and undergoes successive improvements and can be superseded as CMMI has replaced CMM (SEI web site).

The term maturity model is normally associated with process improvement, for example, the testing maturity model (Burnstein *et al.* 1996) and the supply chain management process maturity model (Lockamy III & McCormack, 2004). It seems reasonable that the focus of a maturity model can be either a physical or abstract object, or even an idea that can be developed over time, for example maturity indexing for coal measures (Suggate 2000). Maturity models should provide a set of tools and techniques that include:

1. A set of criteria for characterising the focus of the maturity model.
2. A set of tests for evaluating each of the criteria, often with an associated set of measurement scales.
3. A set of maturity levels that define the stages of growth.
4. A set of goals for each of the levels.
5. A method for ranking the focus of the maturity model.
6. Methods for progressing from one level to the next.

In more sophisticated maturity models the set of factors used to assess the goals for each level would focus on key areas. These would be different for each level so that assessment at one level implies that the key areas at the lower level have been satisfied and those factors characterising the lower level are developed.

4.4 Towards a UI Pattern Language Maturity Model

Updating the set of tests proposed for evaluating the primary and secondary structures of a UI pattern language has resulted in a set of measures that can be used to evaluate the structure of an existing pattern language and determine whether the structure has internal validity. The results can also be used to compare the structure of pattern languages based on the seven attributes that are tested. These two functions are broadly similar to those provided by a maturity model, and with further development they can become the basis for the first stage in the development of a maturity model for UI pattern languages.

The basic set of tools and techniques for the proposed UI pattern language Maturity Model (UMM) are discussed in the following sections.

Characterising criteria

The set of attributes for characterising the structure of a pattern language are those associated with the seven tests discussed earlier: Mapping, Hierarchy, Level, Matching, Complexity, Richness and View.

Tests

The set of seven tests created for assessing the primary and secondary structure of a pattern language form the basis of the proposed maturity model. Each of the tests has an associated five or six point scale against which the criterion being measured can be assessed. The basic units forming the scales are: none, minimal, possibly, developing, developed, and ‘the maturity criterion’. This basic scale has been modified for each test (Appendix A1). The set of tests for assessing the primary structure are aimed at evaluating the internal validity of a pattern language so also define a pass/fail point. The pass requirements for these four tests become the goals for the fully mature pattern language.

Maturity Levels

The proposed set of maturity levels that define the stages of growth are: immature, partially mature, moderately mature and fully mature.

Set of Goals

Attribute Level	Validation Map	Hierarchy	Abstraction	Match	Complexity	Richness	Views
Immature (0)	None Minimal <50%	None Minimal	None Minimal	None Minimal <20%	None / Minimal <5%	None	None
Partially mature (1)	Possibly / Developing <70%	Possibly / Developing	Possibly/ Developing	Developing <=50% match	Possibly / Developing <25%	Possibly / Developing <10%	Possibly / Developing 1
Moderately mature (2)	Developed <90%	Developed 2 levels	Developed	Developed >50% match	Developed <35%	Developed <20%	Developed 2
Fully mature (3)	>=90%	>=3 levels	Modelled	Matched >=80%	Complex >=35%	Rich >=20%	Views >2

Table 4.1 – Maturity level goals for each UI pattern language structural attribute

A set of goals for each of the maturity levels defined for UMM have been defined, and can be found in Table 4.1. The detail for each of the conditions that define the goals can be found in Appendix A1. The goals of the four attributes that define the primary structure of a UI pattern language for the full maturity level are shown as greyed cells in Table 4.1. When these four goals are satisfied then the structure of the UI pattern language is internally valid.

Ranking Method

A single value by which UI pattern languages can be compared and by which progress towards maturity can be measured is calculated by combining the scores allocated to each of the attributes. An attribute is scored as three if its measure meets the goals for a fully mature UI pattern language but a zero if only the immature goals are met. The scores are shown in the parenthesis under the maturity levels shown in Table 4.1.

The maturity rating for a UI pattern language is calculated by first summing the scores for each of the attributes for the four primary structures. Next the sum of the scores for each of the attributes for the three secondary structures is multiplied by four divided by three so that the scores for both primary and secondary of structures have equal weighting. The two values are then added and the result rounded to calculate the rank or maturity index for each UI pattern language. The pattern language maturity scale ranges from the immature language which will have a ranking between zero to six, to the fully mature language that will have a ranking between eighteen and twenty-four.

Maturity Progression

The method proposed for progressing a UI from one level of maturity up to the next is through UI-pattern modelling. Developing UI-pattern models for UIs as part of the UI development process over time will identify deficiencies in a UI pattern language and these can be corrected. This is a generative process of incrementally improvement. It should lead to the restructuring of the pattern language and the addition of new patterns. The result should also improve the pattern language's maturity rating.

4.4.1 Applying the UI-pattern Language Maturity Model

The proposal for the UMM has been used to evaluate the five UI pattern languages used in examples in the preceding Chapter (Sections 3.3 and 3.5).

	Test-one Map	Test-two Hierarchy	Test-three Levels	Test-four Match	Test-five Complexity	Test-six Richness	Test-seven Views
van Welie							
GUI	None	None	None	None	None	None	None
WEB	Developing	Possibly	Possibly	None	Developing	Possibly	Possibly
E-commerce	Mapped	Hierarchical	Developed	Minimal	Developing	Developing	Developed
Borchers							
HCI	Mapped	Hierarchical	Modelled	Matched	Developing	Minimal	Developing
Tidwell							
Common Ground	Possibly	Possibly	None	None	None	None	Possibly
Van Duyne							
Site Design	Mapped	Hierarchical	Developing	Minimal	Developing	Possibly	Views

Table 4.2 - Summary of the pattern languages' maturity model attributes

The results are shown in Table 4.2. The classifications for some attributes are provisional because only subsets of some of the languages have been analysed (Web, Common Ground and Site Design). Borchers HCI language is the only collection that passes all four validation tests. In this collection the links between the patterns were clearly an important consideration when defining the contexts and references of the patterns. The pattern collection is also relatively small so analysis is comparatively straight forward.

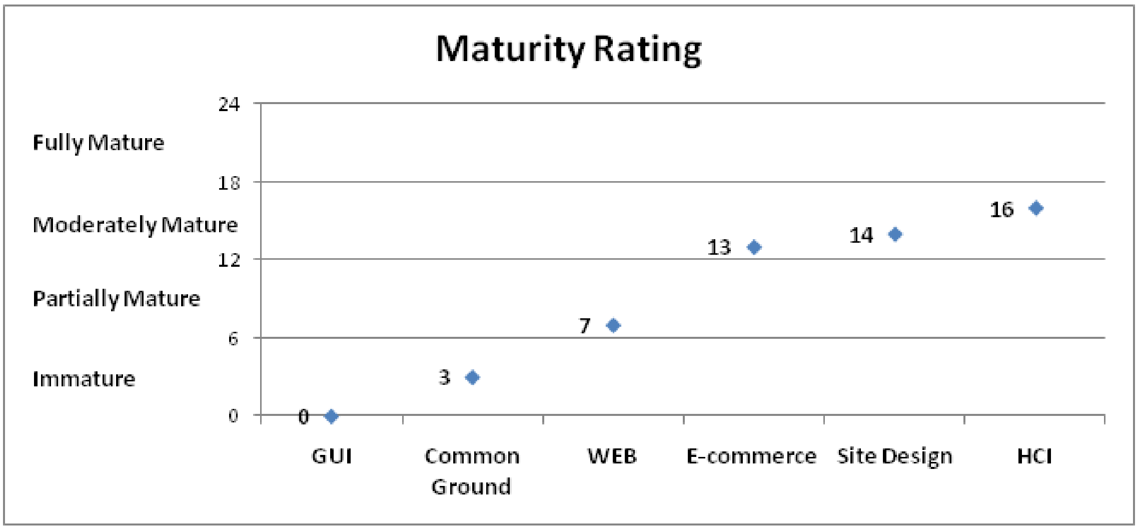


Figure 4.4 – Graph of maturity ranking for each of the UI pattern languages analysed

The goals reached by each UI pattern language for each of the attributes (Table 4.2) can be used to calculate their maturity ratings. These are shown in Figure 4.4. The x-axis has been ordered by the pattern language maturity rating scale. This graph shows that none of the UI pattern languages analysed can yet be rated as fully mature and the GUI language is very immature.

For each level in the proposed UI pattern language maturity model all of the factors have a provisional goal with no key factors associated with a particular level. As the model itself matures it is likely that some factors such as ninety percent connectivity of the validation map will be associated with an earlier maturity level than specified in the current model. For example, connectivity of the validation map could become a key goal for ‘partial maturity’.

UMM can be used to evaluate the maturity status of any collection of patterns therefore the term ‘pattern language’ qualified by its maturity status, can be used rather than alternative nouns such as collection and set. For example the Design of Sites pattern language (van Duyne *et al.* 2003) is a moderately mature pattern language. A pattern language can be characterised by the values assigned to the seven factors assessed by the maturity model. They are: Map, Hierarchy, Levels, Match, Complexity, Richness and Views.

4.5 Discussion – Research Question One

Chapters Three and Four cover issues related to identifying UI pattern language characteristics. The first research question is wholly concerned with characterising a UI pattern language and is discussed below:

Question One: What are the characteristics of a UI-pattern language?

This investigation resulted in the development of four questions to use for evaluating the internal validity of a UI pattern language. Three additional questions identify secondary structures found in a pattern language. The seven attributes that are the focus of these questions are:

1. **Mapped**: Reference and context links between the patterns form a pattern map of basic relationships (contained, is-contained-by, is-a).
2. **Hierarchy**: The pattern map can be organised into one or more overlapping hierarchies.

3. **Levels**: Clearly identifiable levels of abstraction can be used to describe an interface at different levels of granularity.
4. **Match**: Every reference link has a matching context link, except for the root and leaf nodes in a hierarchy.
5. **Complexity**: The number of intra-level links (between patterns at the same level of granularity) and inter-level links (between patterns which are separated by more than one level of granularity) are identified.
6. **Richness**: The number of non-basic relationships between patterns are identified.
7. **Views**: (closely related to hierarchy) A hierarchically structured subset of the patterns that describe the UI from a specific point of view.

Together these attributes characterise any collection of UI patterns but clearly the 'mapped' attribute is the most important as the other six are dependent on it.

The questions used to evaluate these attributes have been incorporated into a UI pattern language maturity model, UMM (Section 4.4). UMM characterises the internal structure of a pattern language. It does not cover external validity or the validity of the individual patterns comprising the language.

UMM provides a method for ranking the maturity status of UI pattern collections. All collections can therefore be referred to as pattern languages. The values assigned to each of the seven attributes can be used to quantify each pattern language's ranking. UMM itself is immature and will be modified with use. For example, the scales of units that the attributes are assessed against will change as more experience is gained. The tests probably require further refinement because they have only been used to evaluate six pattern languages but they are readily usable as has been clearly demonstrated (Table 4.2). Applying the tests is straightforward, but judgement has to be exercised when deciding whether a language meets a specified criterion.

An educator who is not just teaching *about* pattern languages but wants to teach UI design *through* patterns languages (Griffiths & Pemberton 2004), probably requires a UI pattern language with a reasonably developed hierarchal structure. Such a language has the potential to enhance student learning rather than cause confusion as reported in the literature (Dearden *et al.* 2002a, b, Wesson & Cowley 2003, Koukouletsos 2008).

If the six languages (published by: Borchers 2001, Tidwell 1999, van Duyne *et al.* 2003 and van Welie accessed 2003) are a representative sample at the time of evaluation, then most UI pattern languages are not internally valid. On the other hand fifty percent of these pattern languages had an overall maturity status of moderately mature (Figure 4.4). Educators who wish to provide students with real world experiences of the constructivist learning environment (Jonassen 1999), can use this information when evaluating pattern languages for use in teaching. UMM will help identify potential problems students may encounter in understanding and using a pattern language's structure, for example by highlighting omissions and inconsistencies. An internally valid pattern language will have one or more well-developed hierarchical structures identifiable in the validation pattern map. Such information is useful when planning modifications to help scaffold student learning (Wood & Wood 1996, van der Stuyl 2002).

4.6 Conclusion

The set of tests proposed in Chapter Three has been amended to measure the structural attributes of a pattern language. The four attributes defining the primary structure of a pattern language assess its internal validity. The other three attributes relate to its secondary structure. These seven attributes can be used to characterise the internal structure of a UI pattern language in part answering the first of the research questions guiding this research.

The tests assessing the values of the seven attributes' have become the basis for the proposed UI-pattern language maturity model (UMM). As UMM matures how to characterise a UI pattern language should also become clearer since it can be used to evaluate the maturity status of any collection of patterns these can all be called 'pattern languages'. Educators can use the UMM to appraise the structure of existing pattern languages and to identify potential problems students may encounter in using them. They can then use this information when designing teaching material.

The UI-pattern model can be used for representing examples for the purposes of evaluating whether a pattern language passes Test-three. It developed to provide an enhanced graphical representation of the usual sequence (pattern list). A UI-pattern model forms a meta-language which can describe many physical UI designs. The process of revising a pattern language by building a UI-pattern model is also an example of the Alexandrian generative process (Alexander *et al.* 1977). It is a process by which a pattern language is refined by repeated use. Therefore, it has been proposed as the process of incremental improvement which will progress a pattern language through the maturity levels of the UMM. A pattern language should be a ‘living language’ in that both patterns and relationships between patterns can be modified, deleted or new ones added as the language is used to guide the design process. This process of dynamic redevelopment is essential in an area of fast changing technologies.

Chapter 5: Introducing Students to UI Patterns and UI Pattern Languages

The literature review (Chapter 2) identified issues that need further research to determine whether student learning can be improved when using UI patterns. The three main issues (Section 2.4) that appear to influence student learning are:

1. The organisation and structure of the pattern language.
2. The nature of the information presented, including visual aids illustrating different sections in a pattern.
3. The methods used to introduce students to using a pattern language.

The previous chapter investigated issues raised while considering the first research question ("*How can a UI pattern language be characterised?*"). Understanding the characteristics of a well structured pattern language guided the investigation into the structure of a number of existing UI-pattern collections (Tidwell 1999, van Welie & van der Veer 2003, van Duyne *et al.* 2003). Such knowledge is logically a prerequisite when developing techniques to improve students' understanding and use of UI pattern languages.

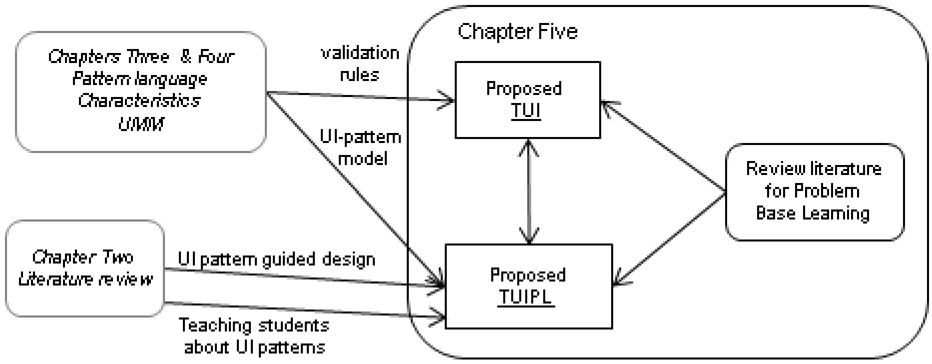


Figure 5.1 - Chapter overview

To identify the characteristics of existing UI pattern languages the investigation explored different ways of using the UI-patterns to represent existing user interfaces. A secondary goal of this exploration is to find a representation that will help students utilise a pattern language's structure to guide UI design. The literature review indentified problems both students and novice UI designers have with using UI pattern languages and particularly highlighting the development of an understanding of the structure of a pattern language (Finlay *et al.* 2002, Kotzé *et al.* 2006, van Biljon *et al.* 2004). These issues relate to the third research question ("*How can student UI* *designersr be successfully guided in the use of a UI pattern language for developing conceptual UI design models?*”). This is the focus of this chapter.

Figure 5.1 shows the inter-relationships between the main themes of the current chapter and its relationships with the preceding chapters. The experience of developing the validation rules, the UMM and the UI-pattern model all provide inputs to this chapter. The chapter starts with a short survey of literature on problem-based learning. Next a method for introducing students to pattern-guided UI design using the UI-pattern model is proposed. As part of this development a set of patterns modified specifically for introducing students to pattern-guided UI design is proposed.

5.1 Problem Based Learning (PBL)

Prince (2004) evaluates active learning (Bonwell & Eison 1991) in the context of engineering education. He defines four instructional methods that are regularly encountered in the literature associated with active learning. These can be used either independently or in combination:

Active Learning should engage students in the learning process by involving them in learning activities that are meaningful to them.

Collaborative Learning emphasises students interacting together as they work together toward some common goal.

Cooperative Learning is similar to collaborative learning in that groups of students normally work together on structured activities but are assessed individually.

Problem-based Learning focuses on the problem which provides the context and the motivation for both instruction and learning.

Although Prince isolated active learning as a specific type of instructional method he also uses the term generally to represent all the other methods. His review identified ‘*extensive and creditable evidence*’ that using these approaches:

1. Results in students becoming engaged in learning with measurable benefits.
2. Improves students’ achievements by a significant magnitude.
3. Promotes collaborative and cooperative learning environments which improve overall student learning.
4. Requires students to become individually responsible and accountable.

5. Promotes positive student attitudes to learning.

In relation to problem-based learning instructional methods, he identified four important outcomes:

1. Traditional test scores are unlikely to improve.
2. Students' attitudes to learning and study habits will improve.
3. Students will retain knowledge for a longer period.
4. Most students will improve their critical thinking and problem-solving skills.

Merrill (2002) reviewed many current educational theories that consider problem-centered instruction including constructivism. He has distilled his analysis of the theories down to five principles that should promote learning when the learner:

1. Becomes engaged in solving real world problems.
2. Builds the new knowledge on a foundation of existing knowledge.
3. Observes a demonstration of the new knowledge.
4. Applies the new knowledge.
5. Integrates the new knowledge into their own world view.

According to Merrill "*these principles are necessary for effective and efficient instruction*" (ibid, p44) and that it is the engagement in 'real' world problems that is key to the other four principles. These identify the four instructional phases of activation (I'm here), demonstration (show me), application (let me) and integration (watch me). Merrill makes the point that he considers collaboration to be an important component of both the demonstration and application phases and not a first principle. He also identifies scaffolding (Wood *et al.* 1976), a form of diminishing coaching, as a significant technique used during the application phase.

The constructivist approach to learning (Jonassen 1999, Jonassen & Carr 2000, Jonassen *et al.* 1996) is normally discussed in the context of using technology as an aid to student learning. But, Jonassen (1999) makes it clear that "*alternative approaches to using technology*" are possible within the constructivist field. Jonassen and Carr (2000) report on how many different types of computer software systems can be used as aids to students learning within the constructivist framework. They suggest that such tools encourage students to develop alternative models or representations of the knowledge they are endeavouring to comprehend. Building knowledge representations is a central foundation of the constructivist approach to learning. Jonassen (1999) defines constructivist learning "*conceptually in an objectivist way*" (ibid, p215) where he clearly describes the methods and instructional activities associated with this approach. He identifies four values which form the foundations of the theory:

1. The problem the students are set to solve is central and should be ill-defined or ill-structured.
2. The student or learner needs to take ownership of the problem (learning goal).
3. Instruction focuses on providing experience that encourages knowledge construction, based on the learner's interpretation.
4. The learner needs to be actively involved and the problem solving involves real world tasks.

Although solving real problems using techniques that professionals actually use is the ideal situation for this form of active learning, providing an 'authentic' experience for the learner is sufficient for quality learning. Jonassen (1999) points out that:

> "*Most educators believe that 'authentic' means that learners should engage in activities which present the same type of cognitive challenge as those in the real world, that is, tasks which replicate the particular activity structures of a context." (ibid, p221)*

The other important feature of the constructivist approach is that worked examples or cases illustrating similar problems and potential solutions should be available. These cases should be graded in difficulty and are part of the scaffolding toolbox. They create a form of surrogate experience and 'memory' of the problem and solution domain that the novice learner can access as required.

Although some theorists suggest that the learner should also identify and define the problem (Bruckman 1998) most imply that the teacher identifies the initial problem but students should then be involved in shaping and defining that problem. Jonassen (1997) identified six attributes of the ideal ill-structured problem:

1. The goals and constraints are unstated.
2. There are multiple candidate solutions or possibly no solution at all.
3. Multiple criteria can be used to evaluate the solutions.

4. There is uncertainty about the knowledge required for solving the problem and that knowledge will normally need organising.

5. There are no general rules or principles that can be used easily to predict the outcome of most cases associated with the solution.

6. The learners will have to make assumptions and judgements about the problem.

Wood and Wood (1996) reviewed the research and theory of tutoring and learning, related to working with individuals, acknowledging that there is a gap between what can be learned about solving a problem alone, and the potential for what can be achieved with guidance or collaboration. Scaffolding provides suitable help when the learner gets into difficulty and less help as the learner becomes more proficient, a form of ‘contingent’ teaching. The goal for scaffolding as a teaching strategy is:

> *“... for the student to become an independent and self-regulating learner and problem solver” (van der Stuyl 2002, p2)*

Wood and Wood (1996) identify four important features pertinent to scaffolding:

1. Tutors need to provide a bridge between existing knowledge and the new knowledge.

2. Tutors need to support a learner’s problem solving efforts.

3. Learners have to become involved in the problem solving experience.

4. Tutors have to aid transfer of responsibility for learning. As the learner begins to solve problems independently, the tutor systematically withdraws their help.

As well, they identify a number of issues related to learning to solve problems. Firstly, the need to learn how to transform a ‘messy’ problem situation into a form which will aid problem solving. Secondly, the need for experience in representing problems using different forms and selecting suitable forms that help solve a problem. Thirdly, the need to develop experience in identifying when something is wrong and learn to self-correct.

5.2 Learning to Use UI Patterns

In the literature review (Section 2.1.3) it was found that a number of authors (e.g. Mahemoff 1999, Borchers 2001a, van Welie 2001, Crabtree *et al.* 2002, Graham 2003, van Duyne *et al.* 2003, Tidwell 2006, Schümmer & Lukosch 2007) make reference to pattern-guided UI design methods. Only one method, Seffah and Gaffer’s (2007) pattern-driven model-based UI development framework (PD-MBUI), provides guidance on how to use the information in UI patterns to create a conceptual model of the required UI. This UI development method is integrated into an automated tool and guidance is via a 'Patterns Wizard' that:

> *"... helps user interface developers in selecting and applying patterns when constructing and transforming the various models to a concrete user interface." (ibid, p1411)*

It appears that most of the methods used UI patterns as a reference source rather than as a 'primary' component within the design process. These methods also assume that some members of the design team will be experienced UI designers who will use many of the solutions captured in UI design patterns (Wania 2008). Experienced designers need only to access UI design patterns for helping solve unfamiliar problems when they are encountered. In general, students are not UI design professionals so do not have a body of knowledge enabling them to access easily the knowledge encapsulated within UI design patterns. Taking an extreme stance, everything will be a new problem for students or trainee UI designers. They need more guidance in how to use patterns effectively.

This observation raises the issue of the type of guidance student UI designers need. Providing guidance can be seen as a form of tutoring or coaching. The problem of UI design is that it is a creative activity where the required UI is often incompletely specified and even when well-specified the requirements usually change during the design process. From the students' perspective, UI design could certainly be classified as an 'ill-structured' or 'messy' problem (Wood & Wood 1996). UI design exhibits the six attributes of an 'ill-structured' problem identified by Jonassen (1999):

1. The goals and constraints for the proposed UI are often incompletely specified and frequently conflicting.
2. There will be many potential UI solutions that could be designed to conform with the specification.
3. UI designs can be evaluated using different evaluation methods and criteria.
4. There is uncertainty about which UI knowledge should be selected for solving a UI design problem. Even when using a tool such as a UI pattern language, that knowledge will normally need re-organising and/or modifying to become useful for a specific UI design problem.

5. Although applying usability rules or principles should improve a design, predicting the outcome of UI designs requires thorough testing with at least a low-fidelity prototype (a physical model).
6. The UI design student will have to make judgements as they develop their UI design and be able to justify their decisions.

Ill-structured problems are central to the constructivist learning environment and UI design can be taught within the framework of this approach. As discussed in the preceding review, encouraging students to create alternative representations of the knowledge by building models is important for helping students to develop their own interpretations. Model building, often in the form of prototyping, is central to UI design. This is normally within the context of a user-focussed UI design method such as Context Design (Beyer & Holtzblatt 1998).

Although the constructivist approach is generally described in the context of students creating their own techniques and modelling representations, an important feature of this approach is that the student experience is authentic. An authentic experience. provides a similar cognitive challenge but is not necessarily the same as real world approaches and tasks. In the world of UI development using formal and informal UI design methods and carrying out modelling tasks are important activities of UI design experts. Most of these methods are not simple 'cookbook recipe' instructions that can be followed mindlessly but require considerable judgement to apply successfully. This can be a significant challenge for many students. Providing scaffolding for these students is important because "*scaffolding focuses on the inherent nature of the task being performed*" (Jonassen 1999, p235).

Scaffolding is central to the constructivist approach to helping students learn. It should be designed to provide temporary aids for the student, especially the less able, so they can become effective learners. As indicated at the beginning of this section, existing pattern-guided UI design methods provide little assistance with how to use patterns once they have been selected from a pattern language. This lack of 'authentic' tasks is a problem for the educator because Jonassen (1999) points out that the teacher

> *"... should determine what kinds of information the learner will need in order to understand the problem" (ibid, p225).*

When teaching UI development his directive implicitly includes identification of suitable UI design methods and UI modelling techniques. It is not clear what to use to help students gain an understanding of using UI patterns during UI design and therefore methods and techniques need to be developed. Jonassen suggests that the teacher:

> *"... should analyze the activity structures required to solve the problems and identify processes that need to be represented visually and how the learner needs to manipulate those images to test their models of the phenomena." (ibid, p226)*

It was in this way that the UI-pattern model described in the previous chapter was developed.

The UI-pattern model has been proposed as a suitable representation for providing evidence of the pattern language passing Test-three of the set of validation tests. During the development of this model the researcher recognised that the UI-pattern model had potential to help students develop their understanding of the structure of a pattern language. Also they could learn about using that structure to select appropriate patterns for describing a UI. This process needs to be incorporated into a detailed UI design method that can be used in the context of teaching UI design.

The structure of the information presented to learners may also influence their learning (Jonassen 1999). It is clear from the review of the literature that the content and presentation of the UI patterns, and the structure of the pattern language are important issues (Section 2.4). Developing a teaching-specific UI pattern language for at least initial instruction accompanied by graded examples is desirable. These would provide a temporary framework supporting students while they learn the tasks and methods, and become familiar with the structure of the knowledge in patterns and the structure of a pattern language.

5.3 Pattern-guided UI Design

In discussing the use of patterns in the UI design process van Welie (2001) says that:

> *"Design patterns capture proven design knowledge and provide means to define a link between user tasks and suitable design solutions." (ibid, p7)*

He assumes that task modelling has been completed and suggests UI design patterns be used as part of the detailed UI design process of which he says

> *"This activity consists of three sub-activities that are strongly interrelated: specifying the functionality, structuring the dialog between the users and the system, and specifying the way the system is presented to the user." (ibid, p4)*

The experience of developing the UI-pattern model suggested that this model could be the basis of a detailed design method bridging the gap between task analysis and the concrete design of the new UI. The PD-MBUI framework (Seffah & Gaffer 2007) does exactly that with the conceptual models being developed in phase two. These UI conceptual models are defined using wireframe diagrams and XUL, with automated UI pattern guidance via the patterns Wizard.

The framework proposed here has been developed primarily with the objective of helping students use UI design patterns rather than for use by UI design professionals, although they are not excluded. The aim is to provide an 'authentic' experience (Jonassen 1999) in as much as the students carry out tasks and are cognitively challenged in ways that are similar to those that the expert UI designer would experience. The method is inspired by Alexander's eight step approach (Alexander *et al.* 1977) but rather than identify the relevant patterns and develop a sequence of patterns, the selected patterns are organised into a UI-pattern model.

The proposed UI conceptual model amalgamates both Constantine's (2003a) canonical abstract prototype diagrams (Figure 2.1) and navigation maps (Constantine 1998). These are combined into a single diagram based on the diagrams created by System Architect (Phillips & Joe 2005). This combination diagram has been named a **T**eaching **U**ser **I**nterface **C**onceptual (TUIC) model. The diagram is used to illustrate the essential elements of a pattern's solution. As stated in Section 2.3.1.2, this model has been chosen to illustrate patterns specifically tailored for teaching students because it is a semi-formal method that uses a clearly-defined minimalist symbol set (Appendix A14). TUIC modelling is similar to using sketching and storyboarding, both techniques typically taught to HCI students, and consequently minimises the learning load for students. The symbols used are visually similar to generalised icons used to represent widgets in an Integrated Development Environment (IDE) toolbox. Both of these advantages conform to the constructivist view that visualisation tools

> *"... must closely mimic the nature of images required to understand the ideas." (Jonassen 1999, p226)*

TUIC modelling has the added advantage that students have to focus on the conceptual requirements of the UI that they are modelling. The students are divorced from the physical toolbox approach of traditional hi-fidelity prototyping. This should encourage them to identify the essential elements required to create a solution rather than identifying physical UI widgets and prematurely designing their layout.

5.3.1 TUIPL: Teaching oriented UI design guided by a Pattern Language framework

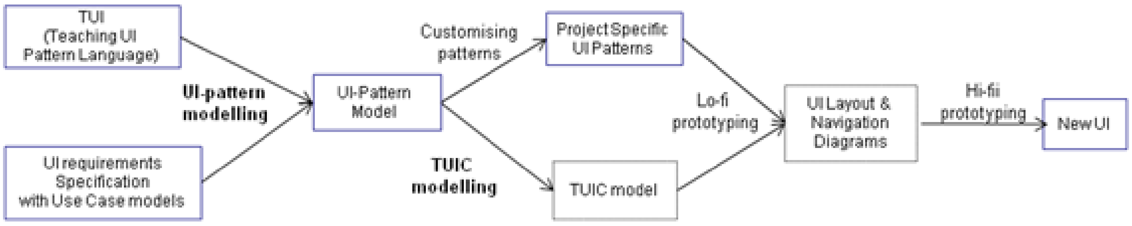


Figure 5.2 - The pattern-guided UI detailed design framework

The **T**eaching **UI** design guided by a **P**attern **L**anguage framework (TUIPL) for the UI detailed design stage is composed of five key processes, two conceptual models, a domain specific pattern language with lo-fidelity and hi-fidelity prototypes as shown in Figure 5.2. TUIPL's five processes are:

UI-pattern modelling takes the information defined in the UI requirements and uses this information to select relevant patterns from TUI. These patterns are then used to create the model by selecting relevant links between the patterns and structuring the graph hierarchically.

TUIC modelling brings together both canonical abstract prototypes (Constantine 2003a) and navigation diagrams (Constantine 1998) to create TUIC model diagrams illustrating the solutions of each pattern making the UI-pattern model. These diagrams are then used to guide the development of the TUIC model for the UI.

Customising patterns takes the information from the UI patterns used in the UI-pattern model and modifies them to refer explicitly to the context of the new UI being modelled. This could include using relevant examples and modifying the TUIC MODEL diagrams to use naming relevant to the UI rather than generalisations.

Lo-fi prototyping produces the UI layout and navigation diagrams, guided by the project-specific UI patterns. The lo-fi prototype is created by restructuring the TUIC model to develop the layout and possibly replacing the CAP symbols with UI toolbox widgets.

Hi-fi prototyping translates the lo-fi prototype into the final user interface.

The whole process is iterative with the insights gained from one activity feeding back to modify earlier outputs and continuing until an acceptable UI is completed.

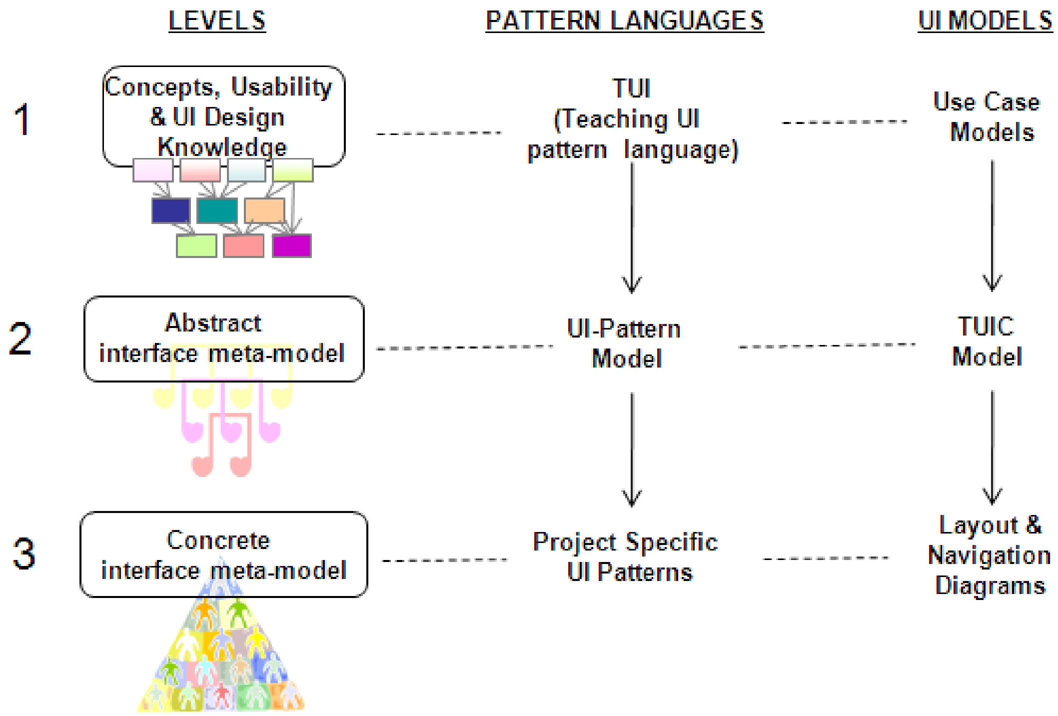


Figure 5.3 – Levels of abstraction matching the pattern languages and models found in TUIPL

The pattern languages developed through following TUIPL are identified in Figure 5.3. In the left-hand column of the diagram are the three levels of pattern language used or created during the UI-design process. The top level depicts the principles, concepts and best practice UI knowledge as embodied in a UI pattern language. Level two is the abstract meta-language formed by the patterns and relationships selected to describe a given UI specification. The third level represents a domain-specific meta-language created by instantiating the patterns forming a domain-specific meta-language reflecting the UI project.

The diagram's middle column represents the UI pattern languages used or created during the pattern-guided UI design method. At the first level is the general UI pattern language, which for the purpose of this research is the TUI pattern language. From this language the patterns that best fulfil the specification of a new user interface are selected. These patterns are linked together to form the UI-pattern model thereby forming an abstract meta-language for the required UI for level two. In the third level a project-specific pattern language is created by modifying the patterns forming the UI pattern model. Seffah and Gaffer (2007) describe this as a process of adaption to the current context of use and defining it as:

> *"A pattern is an abstraction that must be instantiated. Therefore, this step has the pattern P adjusted according to the context of use, resulting in the pattern instance S." (ibid, p1413)*

The generalised text, examples and diagrams of each UI pattern are modified to refer specifically to the UI design project's domain, referencing only relevant links.

Three UI models are also used or developed as part of the TUIPL framework. These are shown in column three. The Use Case model is the expected task modelling method that would be part of the UI requirements documentation at level one. The second level is the conceptual model which in the TUIPL framework is a TUIC model. The third level is represented by the layout and navigation diagrams typical of low-fidelity prototyping.

The TUIPL framework enables the students to become familiar with all three approaches identified by Griffiths and Pemberton (2004) to learn about pattern languages. The building of the UI-pattern model provides an opportunity for students to learn about a UI-pattern language. Using the patterns to learn how to create TUIC models is an example of students learning through using a UI pattern language. Finally the process of customising the UI patterns identified by the UI-pattern model, to reflect the new UI can be used by students as an introduction to discovering and writing their own patterns.

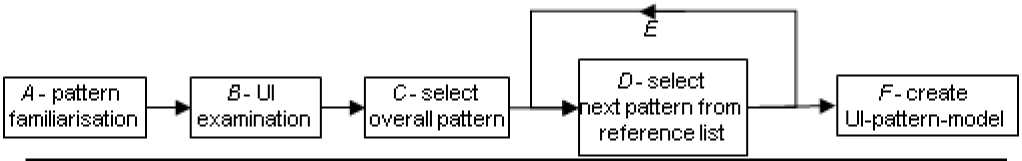
5.3.2 Scaffolding TUIPL

In the previous section the TUIPL framework has been presented. This framework is designed for developing new user interfaces. The goal of a learning unit based on this method would be for students to complete a full UI design project. For the purpose of introducing the students to the UI-pattern modelling method, creating models for an existing user interface is considered more appropriate. It is generally easier to understand an existing UI than to have students become involved in the more intensive creative processes required when constructing models for a new UI. The aim is to focus the students' attention on using UI patterns while learning the new modelling methods. This approach scaffolds the student's initial learning by helping to bridge the gap between having minimal knowledge about the models and methods, and about using them for developing new UI. Re-engineering existing UIs is also a valid ill-structured problem which is regularly carried out in the real world.

Each of the processes in the TUIPL framework poses a difficult problem-solving situation for anyone encountering it for the first time. Learning to create the first two models needs further guidance in how to go about both UI-pattern modelling and TUIC modelling. An argument against providing a step-by-step method for guiding a modelling process is that it restricts students in developing their own independent representations. On the other hand providing a modelling method can also be beneficial as Jonassen (1999) points out:

> *"Modelling tools provide knowledge representation formalisms that constrain the ways learners think about, analyze, and organize phenomena, and they provide an environment for encoding understanding of those phenomena." (ibid, p227)*

5.3.3 UI-Pattern Modelling



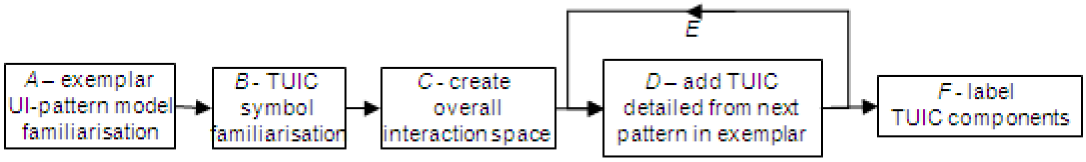
A. Become familiar with the subset of UI patterns.
B. Examine the illustration of the user interface for the exercise.
C. Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information that the UI deals with.
D. From the list of patterns that this pattern references, select those that best describe some other feature of the user interface you have been asked to model.
E. Continue in a similar manner selecting more patterns from the list of patterns referred to at the end of each pattern you select; select patterns that best describe each feature of the user interface until you can't select any more
F. Using the patterns you have selected create a diagram showing those patterns and connecting them with lines to indicate how they are linked into a UI-pattern model.

Figure 5.4 - Method for creating a UI-pattern model when introducing students to UI patterns

The UI-pattern modelling method used to introduce students to UI- pattern languages and using them to model UIs depends on the existence of a suitable set of patterns and an appropriate UI. This method is designed to provide sufficient information for students to create a model but is not so detailed that it does not allow for students to develop their own views of how to go about creating one of these models. This is important because the constructivist approach to PBL encourages students to develop their own representations of the knowledge they are learning.

Figure 5.4 shows the steps in the first version of the proposed method for this process. In this method the students first work through all the patterns in the pattern language to become familiar with them. Next they are required to select patterns that describe the features of an existing user interface. They are directed to use the context and reference links identified in each pattern to limit the search for the next pattern they select. Once they consider they have found sufficient patterns to completely describe the UI they are directed to take this list and organise it into the start of a hierarchical structure by using the link information in the patterns. The resultant linked structure is organised into a UI-pattern model by using only relevant links and duplicating pattern nodes where such duplication improves the depiction of the UI.

5.3.4 TUIC Modelling



A. Find the subset of UI patterns used in the UI-pattern model and become familiar with how they describe the example UI.
B. Become familiar with the TUIC components.
C. Starting with the top pattern, examine the solution and the TUIC model diagram to get an overview of the model you are to create. Draw an outline space and label it.
D. Work down the patterns using the TUIC model diagram to guide your modelling. Use generic symbols for higher level patterns.
E. Use lower level patterns to add CAP details and navigation symbols. Refine generic CAP symbols with specific ones.
F. Name all components on your TUIC model to reflect the example UI.

Figure 5.5 - Method for creating a TUIC model when introducing students to UI patterns

The TUIC model is a form of UI conceptual model and has been selected to encourage students to think at an abstract level about the detailed requirements of the UI they have been asked to model. The method proposed for helping the students develop their models, like that proposed for guiding UI-pattern modelling, is flexible enough for students to have to use their own judgment on how to proceed and how to apply the exemplars provided in each of the patterns. This method also depends on the existence of a suitable pattern language and example UI with a UI-pattern model representing that UI.

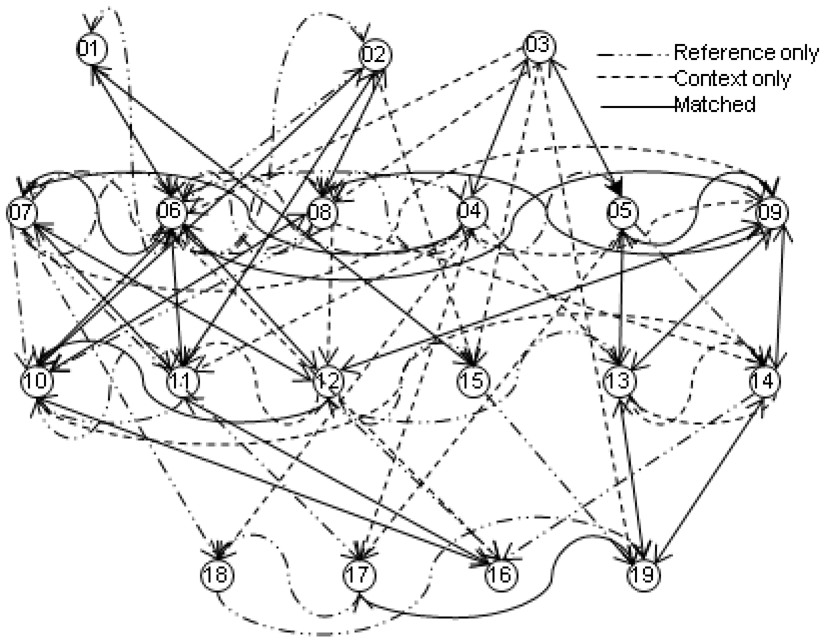
The overview and steps in the proposed TUIC modelling are shown in Figure 5.5. The students are provided with an exemplar UI-pattern model similar to the one they will already have created in a previous exercise. First they need to be familiar with the UI-pattern model and the TUIC symbol list (Appendix A14). Then they use the patterns identified by the UI-pattern model to guide the TUIC model development. The TUIC model diagrams identify the essential elements of the solution in each pattern. Finally the students are directed to update labels used on their diagrams to reflect the domain of the UI they are modelling.

5.4 TUI: The Teaching User Interface Pattern Language

To investigate the impact on student learning of the physical organisation of individual UI patterns and also the inclusion of different items within a pattern, a restricted set of patterns modified for student use is required. Creating a pattern set especially for student use has been found to be beneficial for students in the Software Engineering domain (Freeman *et al.* 2004, Muller 2005) and has since been confirmed in the UI Web design domain by Koukouletsos *et al.* (2007). But there is a problem of how far a pattern language specifically designed for students needs to diverge from those found in the real world.

The proposed pattern-guided UI design method requires a language that has sufficient frequency of the three basic relationships (Section 4.1.1) so that a UI-pattern model can be created. For expert UI designers this requirement is probably not necessary because they should have enough knowledge to readily recognise missing relationships and as a consequence could update any pattern collection they are using. Most students will not have this knowledge and as the literature (Finlay *et al.* 2002, Wesson & Cowley 2003, Kotzé *et al.* 2006) indicates they would have difficulty using a collection that is poorly structured.

The set of patterns that had most often been used in earlier investigations (see Tables 2.2 and 2.3) was Tidwell’s Common Ground collection (1998, 1999). Therefore this set has been used as the basis for the first version of TUI. The analysis of three UI examples that had been successfully used in previous HCI courses at Massey University has been used to determine a suitable subset of patterns for TUI. Partial pattern maps were created for three UI examples. The narrative content of the patterns is based on Tidwell’s Common Ground (1998, 1999). Where an appropriate pattern could not be found a pattern from Tidwell’s (2006) version of the pattern language has been added. One additional pattern ‘07 Master with Details’, has been based on Bradac and Fletcher’s (1998) Form Style patterns.



01 – Narrative	11 - Tabular Set
02 - High-density Information	12 - Stack of Working Surfaces
03 - State Information	13 - Choice from a Small Set
04 - Transaction by Form	14 - Choice from a Large Set
05 - Control Panel	15 - Convenient Environment Actions
06 - Collection beside Content	16 - Pointer Shows Affordance
07 - Master with Details	17 - Forgiving Text Entry
08 - Cascading Collections	18 - Input Prompt
09 - Groups with Titles	19 - Good Defaults
10 - Hierarchical Set	

Figure 5.6 - Pattern map for initial version of TUI

Figure 5.6 shows the pattern map for the nineteen patterns making up the first version of TUI. It would have been possible to modify the context and reference sections of the original patterns so that the resultant pattern map was internally valid (Section 4.1.1). This may assist student learning because research had already identified that they had difficulty understanding the relationships between patterns (e.g. Kotzé *et al.* 2006). After due consideration, it has been decided to use a realistic pattern map.

Changes have been made to the patterns so that UI-pattern models could be created for each of the three examples. However some links are only mentioned in either the reference or the context section of the related patterns. Additional patterns have been added so students have to make choices, for example between the patterns '10 - Hierarchical Set', '11 - Tabular Set' and '12 - Stack of Working Surfaces'. The pattern language has enough patterns to enable UI-pattern models to be created for simple user interfaces but pattern selection is limited to ensure that only single hierarchies can be created.

Attributes	Map	Hierarchy	Levels	Match	Complexity	Richness	Views
TUI	mapped	hierarchical	modelled	developing	possibly	developing	possibly

Table 5.1 - Values of TUI's maturity model attributes

TUI's structure has been assessed to check that it is realistic by using UMM. The values assigned to each of the attributes are shown in Table 5.1. All patterns in TUI link into the validation pattern map, and this map can be organised hierarchically as shown in Figure 5.6. The levels can be used to describe a UI at different levels of granularity and a number of UI-pattern models have been created using it (Appendices A5 & A7). Most relationships in the context and reference sections are basic relationship types but only thirty-four percent of them match. There are three potential root patterns in TUI which can be used for defining the overall context for a UI. Most of the patterns relate to each other spatially but some with temporal relationships have been included. The UIs that can be described using TUI are basic screen designs. The maturity model classification of 'possibly' has been assigned indicating that partial hierarchies can be created.

Over fifty percent of the links in the resulting graph are intra-level links or inter-level links. So although this UI pattern language is small it is very complex. This complexity is partially due to the language being predominately a subset of an existing pattern language with a low maturity rating (Figure 4.4). Some alternative relationships have also been retained so that richness is just developing. The rationale for leaving this complex structure in TUI was the similarity of the structure to both van Duyne *et al.*'s (2003) Design of Sites pattern language and van Welie's 'Patterns in Interaction Design pattern language (accessed 2003). Both have been used in a number of the studies (Wesson & Cowley 2003, Kotzé *et al.* 2006, Segerstahl & Jokela 2006). This first version of TUI is moderately mature with a maturity rating of 14 which is similar to that of van Duyne *et al.*'s (14) and van Welie's e-commerce pattern language (13).

A major influence on the decision about the link structure of TUI is the premise that students would be able to work with a realistic pattern language structure because they would be introduced to it through the construction of a UI-pattern model. This approach in part conforms to the constructivist point of view that the type of problem students ought to solve should be ill-structured or messy, and authentic. The structure of TUI has been progressively modified as the research progressed.

Several studies (Finlay *et al.* 2002, Dearden *et al.* 2002a, Chung *et al.* 2004, Segerstahl & Jokela 2006) reported that their subjects identified the illustrations as being an important feature of the patterns. Illustrations are known to be important for enhancing learning. Carney and Levin (2002) review the research about the impact of pictorial illustrations on student learning of the accompanying text. They identify many ways in which illustrations can help students learn textual material. These include: increasing motivation to read the text, focusing the learner's attention, helping to clarify content and serving as mental models. They also identified five different types of illustration. The purely *decorative* illustration will have little or possible even a negative effect on learning. The example they give is a stylised sketch of a pine tree with text about a tramping track. The next type of illustration is the most common; a picture depicting some aspect of the text it accompanies, including examples. These *representational* pictures can have a moderate impact on student learning. The next two types of illustration have more impact on learning. The *organisational* picture provides structure for the text, as a map would for text about a tramping track or a diagram highlighting the main steps in a process. The *interpretational* illustration is the type of diagram that clarifies some difficult part of the text. The type of illustration that can provide the most benefit for learning is called *transformational*. These illustrations can also be the most difficult to design. They "*include systematic mnemonic components that are designed to improve a reader's recall of text information*" (Carney & Levin 2002, p7). There is a recognisable visual element related in some way to the title of the text and it recurs in a series of illustrations related to different parts of the text.

Some researchers (e.g. Dearden *et al.* 2002a) cautioned that there is a tendency for students to rely on the illustrations when making UI design decisions. No studies could be located that actually tested whether this warning had any foundation in reality. Three versions of TUI were used to investigate the role of illustrations when introducing students to a UI pattern language.

1. The narrative pattern type - where the patterns have no illustrations.

2. The illustrated pattern type - where the patterns have a large screen shot illustrating a typical example of the UI issue the pattern addressed. These illustrations can be classified as representational.

3. The diagrammed pattern type - where the patterns include a diagram to illustrate the essential elements of the solution. These diagrams can be classified as interpretational. In these patterns all the screen shots illustrating the examples were placed at the end of the pattern.

5.5 Summary

A pattern-guided UI design framework TUIPL has been proposed for the detailed UI design process of UI development. Learning to design new artefacts like UIs is intrinsically a process of problem-based learning. TUIPL has been developed in the context of the constructivist approach, a form of problem based learning. It has been designed primarily for teaching UI design students about using UI patterns to design UIs, with the aim of helping students learn about UI pattern languages and how they might be used in UI design. The framework is compatible with existing user-centered techniques and is not precluded from being used as part of a commercial UI development project. It should provide students with an authentic experience, a necessary goal of creating a constructivist learning environment. This method can be used to introduce students to the development of conceptual UI models in the form of TUIC models.

Finally a pattern language, TUI containing nineteen patterns has been created. TUI has been designed to be used by students with TUIPL. The first version of TUI described in this chapter is not internally valid and its maturity is comparable with commonly available pattern languages at the time the initial version was created (2006). This complexity is partly due to the language being primarily a subset of an existing immature pattern language. A realistic UI pattern language structure has been developed for TUI, so that an ‘authentic’ situation can be provided for students, in-line with constructivist thinking. It has been surmised that by constructing a UI-pattern model, students will be able to develop their understanding of the structure of a UI pattern language, overcoming the difficulties reported in the literature.

This research is focussed on the first two processes in TUIPL, from UI requirements to the development of a TUIC model. The first process is the UI-pattern modelling method and the second is the TUIC modelling method. The investigations studying these two processes are reported in the following chapters.

Chapter 6: Research Overview

Three studies have been designed to address the issues raised by the literature review and the outcomes reported in Chapters Three, Four and Five. These studies are carried out sequentially with the results from one study leading into the next.

The first part of this chapter describes the approaches used. The three studies are then introduced. The second part of the chapter describes the different versions of the experimental artefacts. Finally, there is a description of the processes each of the three studies use. These have been placed in this chapter to reduce duplication because they have many features in common. Consequently, they are more easily compared.

6.1 Research Questions

The three questions that this research addresses are restated here:

1. What are the characteristics of a UI-pattern language?
2. Are there any specific requirements of a UI pattern language when used for teaching student UI designers?
3. How can student UI designers be successfully guided in the use of a UI pattern language for developing conceptual UI design models?

The research is structured around three main threads. These are: pattern language characteristics, pattern form and UI pattern-informed design.

The first thread, reported in Chapters 3 and 4, explored the characterisation of UI pattern languages. The attributes identified describe the structure of UI pattern languages and the investigation resulted in the development of UMM (Section 4.4). UMM can be used to characterise pattern languages and compare and contrast them. The application of UMM to evaluate existing pattern languages provided the inspiration for the development of the UI-pattern model. This model has been recognised as having potential for scaffolding students' understanding of the structure of a pattern language.

The second thread is to ascertain whether UI patterns linked into a pattern language are a suitable tool to use with UI design students. The literature review identified the benefits of using UI pattern languages with students and groups of designers to inform UI design but also identified a number of issues that need addressing (Section 2.2.3). The visual elements used to illustrate patterns have been identified as being important (Finlay *et al.* 2002, Dearden *et al.* 2002a, Chung *et al.* 2004, Segerstahl & Jokela 2006) but when included in UI patterns users may place too much emphasis on them (Dearden *et al.* 2002b). No studies could be found that tested whether this caution had any foundation. This investigation considers whether the pattern form used has any consequences for student learning focussing on the types, position and size of the illustrations.

The impact of pattern language structure on student learning is also explored because several studies report that new users of pattern languages have difficulty using the structure of a pattern language (Finlay *et al.* 2002, Wesson & Cowley 2003, Kotzé *et al.* 2006, Koukouletsos 2008).

A number of studies reported that students' retention of UI design knowledge (Borchers 2002) and that the UI models they created improved (Laakso *et al.* 2000) after being introduced to using UI patterns. Koukouletsos *et al.* (2009) is the only study that confirmed that students had no prior knowledge before being introduced to a set of UI patterns covering that knowledge. They report that the students developed UI prototypes that applied the usability principles introduced in the UI patterns.

The third thread investigates the use of TUIPL as a suitable UI design framework for students follow when learning pattern-guided UI conceptual modelling. The premise is that as a consequence of using TUIPL students should develop an understanding of UI patterns and UI pattern languages. Students should also learn the UI knowledge contained within the patterns (Seffah 2003, Borchers 2002, Koukouletsos *et al.* 2009). The constructivist approach to learning (Jonassen 1999) influenced the development of TUIPL. Constructivist learning should encourage students to use problem transformation to develop an understanding of UI patterns, UI pattern languages and detailed UI design by model building. Students should be provided with a UI design experience similar to a real world experience when applying TUIPL. In the second study the UI-pattern model and the diagrammed version of TUI with TUIC model diagrams illustrating the solution of each pattern should provide scaffolding (Wood & Wood 1999) to guide the students in building a TUIC model.

6.2 Research Approach

The over-arching research method used for this research is best described as an exploratory "Design Study" (Brown 1992, Cobb *et al.* 2003, Shavelson *et al.* 2003, Bell 2004, Collins *et al.* 2004). This approach is used where the research is taking place in an educational setting as part of the normal instructional programme.

> *"Design studies are iterative in that they involve tightly linked design-analysis-redesign cycles that move toward both learning and activity or artifact improvement. They are process focused in that they seek to trace both an individual's (or group's or school system's …) learning by understanding successive patterns in the reasoning and thinking displayed and the impact of instructional artifacts on that reasoning and learning." (Shavelson et al. 2003, p26)*

Design Studies typically use multiple data collection methods and multiple research methods over multiple iterative cycles, often extending over a long period of time. In the early exploratory stages the type of issue addressed is of the form "*What is happening*?" (Shavelson *et al.* 2003, p28) This type of enquiry is answered using data collection and research methods that tend to be descriptive in nature. The second type of issue addressed is of the form "*Is there a systematic effect*? (Shavelson *et al.* 2003, p28) This type of enquiry is used to establish whether there might be a causal relationship, for example between the use of a teaching artefact and student learning. Shavelson *et al.* (2003) suggest that quasi-experiments can often be used to answer this type of question. The third type of issue can only be addressed by a mature Design Study during the latter iterations. By this stage, the research should be systematically identifying causal relationships and mechanisms. The answers sought focus on queries of the form "*Why or how is it happening*?"

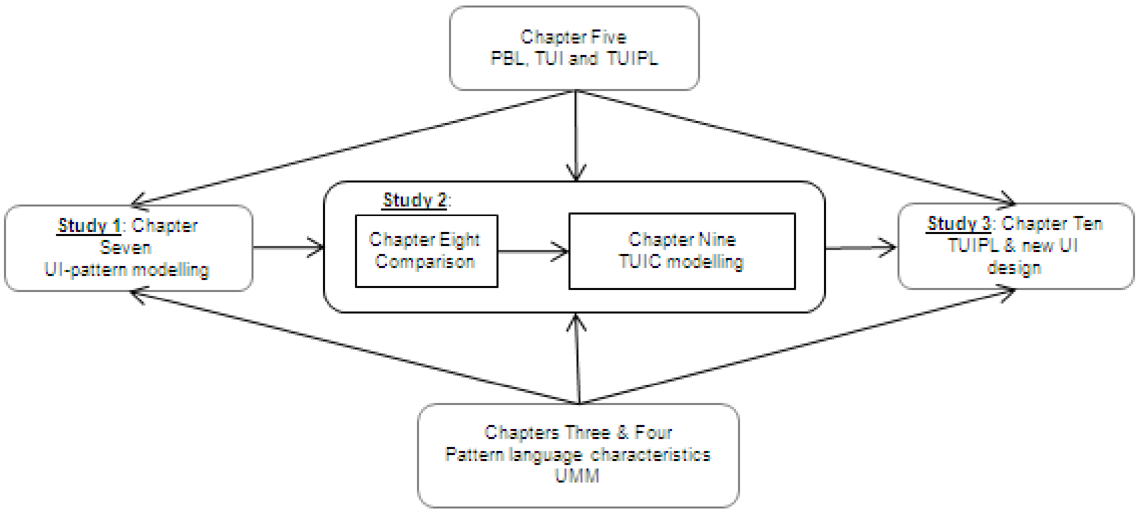


Figure 6.1 - Interrelationships between the different parts of the research

Chapters Three, Four were focused on answering the first research question. Chapter Five introduces the learning approach, TUI and TUIPL. Figure 6.1 shows that the findings from all three of these chapters are inputs for the sequence of three studies that address the last two research questions. Each of the studies uses a multi-methodology research design (Mingers 2001) in combination with different methods for data collection and analysis. An advantage of using this approach is that the different methods can "*focus attention on different aspects of the situation*" (Mingers 2001, p243) another advantage is that triangulation (Benbasat *et al.* 1987, Yin 1994, Mingers 2001, Kauloi *et al.* 1998) can be used to validate the results. Triangulation is "*the explicit use of multiple methods, measures and approaches*" (Wilson 2006, p46). It aims to confirm a finding from one source by matching it to a similar finding from another source thus adding support to conclusions (Benbasat *et al.* 1987, Yin 1994).

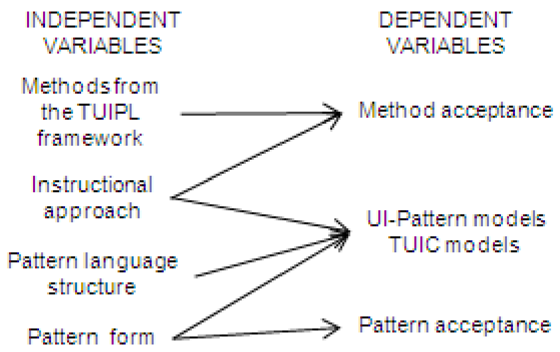


Figure 6.2 - Relationships between dependent and independent variables

As previously stated, this research is exploratory and at the beginning stages of a Design Study. The research questions are all of the first form identified by (Shavelson *et al.* (2003) with some tendency towards his second form of question. For example, the results from the two iterations of the first exercise (trialled in Studies One and Two) are used to compare the narrative and illustrated forms of TUI, effectively a multi-group quasi-experiment (Alvarez *et al.* 2006). A pre-experimental approach (Adelman 1991) is used to trial the TUIC modelling method in Exercise Two of Study Two. Finally, the first two processes in TUIPL are evaluated by three case studies in Study Three where models for a new UI are created. The quality of the UI-pattern models and TUIC models created by the participants in the different studies is a surrogate measure of the success of TUIPL. Results based on a surrogate need to be confirmed from other sources.

> *"Because the surrogate is an indirect measure, there is danger that a change in the surrogate is not the same as a change in the original factor." (Pfleeger 1994, p17)*

Figure 6.2 shows the relationships between the dependent and independent variables investigated in Studies One and Two. Several different data collection methods are used and the results of analyses are triangulated to confirm findings about the dependent variables being investigated. All three studies in this research are introduced sequentially in the following sections.

6.2.1 Study One Overview

This study is to assess the proposed UI-pattern modelling method, the first process in TUIPL. It is a quasi-experiment, chosen because:

1. The Design Study is in its early stages.
2. It is highly desirable to embed the experiment into an existing course. Therefore the experimental process needed to be as similar to a conventional tutorial format as possible.
3. A controlled experimental design with randomly allocated groups could not be devised due to insufficient class time being available. It was not possible to guarantee that no student would be disadvantaged by being exposed to different treatments. Also, when carrying out research within the classroom setting with small classes it is not feasible to select statistically valid random samples.
4. The investigation includes the application of the proposed UI design modelling method. A search of the literature associated with designing experiments to evaluate different methods and processes for designing software identified the premise that there is no way to effectively administer a "placebo" to a control group. Pfleeger (1995b) pointed out that:

> "*If we tell designers not to use a particular method, they are likely to use an alternative method rather than no method at all. The alternative method may be hidden, based on how they were trained or what experience they have had …" (ibid, p 14)*

Two versions of TUI are compared, the 'narrative' pattern form and the 'illustrated' pattern form. The comparison of the two pattern forms, and any generalisations based on the results may have their validity threatened by the lack of random sampling and absence of a control group. To limit such risks to validity, multiple data sources are used (Benbasat *et al.* 1987, Pfleeger 2005) so that results can be triangulated (Wilson 2006, Yin 1994).

<table>
<tr><td colspan="2">Research Aim: Explore the use of a user interface pattern language in the creation of user interface pattern models by student UI developers.</td></tr>
<tr><td colspan="2">Objectives:
⇒ Determine whether UI patterns are an acceptable medium for presenting UI information to students.
⇒ Investigate whether illustrated examples influence student understanding of patterns and pattern languages.
⇒ Determine whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model.
⇒ Discover whether building a UI-pattern model develops student understanding of UI patterns and pattern language structure.</td></tr>
<tr><td colspan="2">Participants: Fourteen third year undergraduate students enrolled in a first HCI course Massey University, NZ, during May, Year 1</td></tr>
<tr><td colspan="2">Research method: – Design study primarily a quasi-experiment
Within-subjects design was used to investigate the second objective for this study.
Matched-pairs sampling, students allocated to one of two groups randomly.</td></tr>
<tr><td>Components of the Study</td><td>Data collection</td></tr>
<tr><td>Introduction</td><td>Exit Questionnaire – section A</td></tr>
<tr><td>Demonstration</td><td></td></tr>
<tr><td>Using narrative and illustrated UI-patterns</td><td>Patterns Questionnaire
Exit Questionnaire – section B</td></tr>
<tr><td>UI-pattern modelling</td><td>Observation – images
Resultant UI-pattern models
Exit Questionnaire – section C</td></tr>
<tr><td>Communication aid</td><td>Observation – images
Exit Questionnaire – section E</td></tr>
</table>

Table 6.1 - Overview of Study One

Table 6.1 provides an overview of Study One showing the guiding aim and associated objectives. It identifies the data collection methods used for each part of the study. Data collection is both qualitative and quantitative. To compare the narrative and illustrated versions of TUI, a within-subjects design is used.

The population from which the student sample is drawn consists of all those students who are likely to enrol in the introductory HCI course over the years that the course is taught. The sample consists of all the students enrolled in the course in a specific year. Students are first organised into matching pairs based on past performance in an HCI-focussed assignment. Each member of the pair is randomly allocated to one of the two treatment groups creating equivalent groups. Threats to external validity (Alderman 1991, Kitchenham & Pickard 1998) are reduced because the students are learning in their natural setting of a standard tutorial within the course.

Observational data is primarily in the form of a series of digital photographs of the workspace used by each pair of students. These are analysed using description, pattern matching and simple counting. The scores obtained from assessing the resultant UI-pattern models are compared using Student's T-test. This is a robust test for small samples drawn from populations that approximate a continuous normal distribution. A robust test is essential because of the potential threats to internal validity posed by a sample that is inherently self-selected because the students chose to enrol in the HCI course (Alderman 1991, Rosenshine 1970).

The students' responses to the Patterns Questionnaire are cross-correlated to assess whether students answered independently, having worked in pairs. Counts and percentages are used where appropriate. The data describing the students' opinions of using the different pattern forms is tested using CHI-square, robust test for comparing distributions. Explanatory narrative (Gerring 2004, Flyvbjerg 2006) and descriptive analysis (Ragan *et al.* 2004, Shavelson *et al.* 2003) are used with the Exit Questionnaire responses with the addition of some counts. For the questions that required a response on a five point scale, descriptive statistics are used.

6.2.2 Study Two Overview

Study Two is the second iteration in this Design Study and has two parts. The first part implements relevant findings from Study One. The protocol from the first exercise in Study One is re-used for Exercise One in this study.

Research Aim: Explore the use of a user interface pattern language in the creation of conceptual user interface models by student UI developers.	
Objectives: ⇨ Confirm that UI patterns are an acceptable medium for presenting UI information to students. ⇨ Confirm that the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model. ⇨ Discover whether implementing the findings from Study One helped students build better UI-pattern models and helped develop their understanding of pattern language structure. ⇨ Determine whether the TUIC modelling method from TUIPL will support students in using UI patterns as a guide to the creation of TUIC models. ⇨ Determine whether UI patterns can be used by student UI designers to gain new UI knowledge.	
Participants: Twenty-four third year undergraduate students enrolled in a first HCI course Massey University, NZ, during May year 2	
Research method: - Design study iteration 2 **Part One** Copied Study One, Exercise One with improved teaching materials – comparing results from the two iterations **Part Two** Trials learning TUIC modelling from UI-patterns	
components of the Study	**Data collection**
Introduction	Exit Questionnaire – section A
Demonstration	
Using **diagrammed** UI-patterns	Patterns Questionnaire Exit Questionnaire – section B
UI-pattern modelling	Observation – images UI-pattern models assessment Exit Questionnaire – section C
Demonstration	
TIOC modelling	Observation – images UI-pattern models assessment Exit Questionnaire – sections B & D
TUIC test	Test assessment
Communication aid	Observation – images Exit Questionnaire – section E

Table 6.2 - Overview of Study Two

Initial analysis of the data from Exercise One is mostly descriptive as for a pre-experiment, using a similar approach to that used in Study One. This analysis is to check that the changes that implemented findings from Study One do not generate any anomalies in student learning. Observational data, primarily photographic images, is analysed using description, pattern matching and simple counting. The scores obtained from assessing the resultant UI-pattern models are presented using descriptive statistics.

The students' responses to the Patterns Questionnaire are presented using counts and percentages. Only the results from Section C in the Exit Questionnaire are relevant to this part of the study.

Table 6.2 provides an overview of Study Two showing the guiding aim and associated objectives. It also identifies the data collection methods. Data collection is both qualitative and quantitative. A quasi-experiment approach is used to compare the results of the first exercises from both studies. A recognised flaw in the experimental design is that more than one of the independent variables has been changed. Ideally, Exercise One should have been repeated, making only one change per repeat. Because the experiment can be run only once per year this is impracticable. A second flaw common to many design studies that trace improvement in instructional artefacts over a number of iterations (Cobb *et al.* 2003) is where the two groups being compared are drawn from cohorts of students from successive years.

To compare the results similar analyses to those for Study One will be used. But in this study three independent variables are of interest: the instructional approach, the pattern form and the pattern language structure. The UI-pattern models the students create depend on all three, consequently results from multiple data sources are triangulated to reduce threats to the validity of the conclusions (Kitchenham *et al.* 2002b).

The second part of Study Two is an exploratory trial or a post-test pre-experiment (Adelman 1991) to discover whether students could learn new UI knowledge from working with patterns and to test the viability of the proposed method for TUIC modelling. The protocol used is a modified version of that used for the first exercise. The data collection methods are also similar but a TUIC test is applied as part of the assessment of students' knowledge of the TUIC modelling. No pre-test is used in this pre-experiment because firstly, knowledge of the student body indicates that students have no prior knowledge of TUIC modelling. Secondly, a pre-test would alert students to the technique which may influence the degree of attention they give to the TUIC model diagrams used in the patterns. Thirdly, there is not enough time in the tutorial session to administer a pre-test. Analysis is through narrative description with the use of descriptive statistics, counts and percentages where appropriate.

6.2.3 Study Three Overview

The research approach applied in this study is primarily a multiple case study where the processes and artefacts are used so that each case replicates the other (Galliers 1992, Yin 1994, Mingers 2001, Flyvbjerg 2006). The approach used incorporates some aspects of descriptive/interpretive research (Galliers 1992) as the cases are partly analysed using description and due to the involvement of the researcher as a participant-observer (Galliers 1992, Yin 1994) there is an inherent bias in the interpretations. It takes the form of three 'descriptive case studies' (Galliers 1992, Yin 1994, Mingers 2001, Flyvbjerg 2006). The case studies explore the viability of TUIPL for designing new UI-conceptual models and whether TUIPL has potential for use in a professional UI design setting. It is not yet appropriate to evaluate TUIPL in the real world. The current goal is to provide students with an 'authentic' UI design experience (Jonassen 1999) not to develop a commercial UI design methodology. Although case studies normally investigate individuals, groups or populations of people in a natural setting the unit of study in a case study can be many other things, such as an event, a program, a time period or a critical incident (Patton 1980). The unit of study for these case studies is the first two processes in the TUIPL framework (Figure 5.2).

The strategy used for identifying suitable cases is one of "maximum variation cases" (Flyvbjerg 2006) where the three cases are selected so that they are different in at least one dimension (e.g. UI design experience). This strategy helps reduce the threat to the validity of generalised conclusions. Generalised conclusions from case studies are difficult as external validity can easily be compromised. Trochim (2006) identifies three major threats to external validity: people, places and time. He suggests that to overcome such threats, studies should be pursued in a variety of places, at different times and with different people. Although, case studies are normally carried out in a natural setting, where and when the exercises are completed in this research is not relevant because the unit of study is the first two processes in TUIPL with the associated models. The three case studies are carried out in the same meeting room but the three developers have varying years and types of UI development experience.

Each participant trialled TUIPL in an artificial setting, solving a simulated problem. Therefore they are effectively role playing at being UI designers. There is a participant-observation (Gerring 2004, Kaplan & Duchon 1988, Yin 1994) component because the researcher is both observing and teaching each participant about the new method and providing help when requested. The participants have to learn the two modelling methods and then apply them by building models for the new UI. Data is collected from multiple sources for each case study: by questionnaires, informal discussion and scoring the resultant models.

Research Aim: Explore whether the TUIPL framework will provide students with an authentic UI design experience.	
Objectives: ⇒ Determine whether UI developers could learn and apply the first two phases of the TUIPL framework in the development of a conceptual UI model for a new user interface. ⇒ Determine UI developers' *opinions of the viability of* the first two phases of the TUIPL framework in a *professional* setting.	
Participants: Three UI developers	
Research method: Case study Multiple descriptive case study with the unit of study being the pattern-driven UI design method Within and between case analysis is used.	
Components of the Study	**Data collection**
Introduction	Exit Questionnaire – section A
Demonstration	
Using **diagrammed** UI-patterns	Patterns Questionnaire Exit Questionnaire – section B
UI-pattern modelling	Observation – images UI-pattern models assessment Exit Questionnaire – section C
Follow-up	Open ended discussion
Demonstration	
TUIC modelling	Observation – images TUIC models assessment Exit Questionnaire – section D
CAP test	Test assessment
Opinion communication aid	Exit Questionnaire – section E
Opinion aiding non-professional UI design team members	Exit Questionnaire – section F
Follow-up	Open ended discussion

Table 6.3 - Overview of Study Three

Data collection methods are similar to those used for Studies One and Two as shown in Table 6.3. The main differences are first, that observations are by the researcher and do not include a series of photographs. Secondly, a short follow-up discussion concludes each of the two main modelling sessions to allow the participant to comment freely on any aspect of the exercise and for the researcher to clarify any points of uncertainty. The Case Studies are divided into three sessions and appropriate sections of the Exit Questionnaire are administered at the end of each session.

6.3 Data Collection

To overcome some of the limitations of the research design, data was collected using different methods, both quantitative and qualitative. Eight different types of data collection technique are used over the course of the three studies (S1, S2 & S3):

1. Observational data (Benbasat *et al.* 1987, Yin 1994) using a sequence of digital photographs taken at regular time intervals. (S1 & S2)

2. Observations (Patton 1990, Yin 1994) are made by the researcher throughout all three of these studies. (S1, S2, & S3)

3. Model scores are used as a surrogate for success of the methods (Pfleeger 1994, Kitchenham *et al.* 1994). Two types of model (UI-pattern models and TUIC models) are created by participants. These are scored using marks for a number of

categories that check for appropriate usage and also for how closely participants' models match the exemplar. (S1, S2, & S3)

4. Participants' perceptions of the UI-patterns and their use are elicited via the Patterns Questionnaire (Pfleeger &Kitchenham 2002, Kitchenham & Pfleeger 2002a). The data collected is in the form of a three point scale: 'agree', 'undecided' and 'disagree'. (S1, S2, & S3)

5. Participants' knowledge of CAP and navigation components used to create the TUIC models are evaluated using a short test. (S2)

6. Open-ended questions (Ragan *et al.* 2004, Yin 1994) asking for participants' perceptions and opinions on the methods and tools used in the modelling exercises are used in most sections of the Exit Questionnaire. (S1, S2, & S3)

7. Close-ended questions (Pfleeger & Kitchenham 2002) requiring participants to respond on a five point scale, from unhelpful to extremely helpful, are included in the Exit Questionnaire (S1, S2, & S3)

8. Additional comments and opinions are elicited through informal discussion (Patton 1990) to confirm ad hoc observations by the researcher. (S3)

As far as possible data collection for Studies One and Two should be carried out without disrupting the standard teaching tutorial environment. Neither close observation by the researcher nor videoing each pair of students as they completed the modelling exercises is practical. Instead, digital photographs of the work surfaces (desk tops) are taken at regular intervals. More details about data collection methods are found in the appropriate chapters describing each of the studies and associated appendices.

6.4 Ethical Considerations

For all studies the Massey University Ethics Approval process is followed. The results from working through the screening questionnaire[4] indicate that none of the three studies require full ethical approval. A "Low Risk Notification" has been lodged in the university's Low Risk Database for each of these studies.

An information sheet explaining the research accompanies the ethics form participants are asked to complete for each of the studies (Appendix A4)

[4] http://humanethics.massey.ac.nz/massey/research/ethics/human-ethics/approval.cfm

6.5 Research Participants

For Studies One and Two the participants are undergraduate students enrolled in the third year introductory HCI paper at Massey University. All students in both classes used for the studies agreed to participate. The experiments are embedded within the normal tutorial program.

Study Three is a set of case studies investigating the TUIPL framework. The participants represent user interface developers with a range of commercial experience.

6.6 Pattern-guided UI Design

Chapter Five briefly examined the literature relevant to Engineering Education, Active Learning, PBL and Constructivism. The requirements for developing suitable tools and techniques to effectively scaffold student UI designers in learning about using patterns in UI design have been proposed. These focus on the TUIPL framework. For teaching, TUIPL uses the TUI pattern language. Chapter Five concluded with a set of requirements for the initial version of TUI.

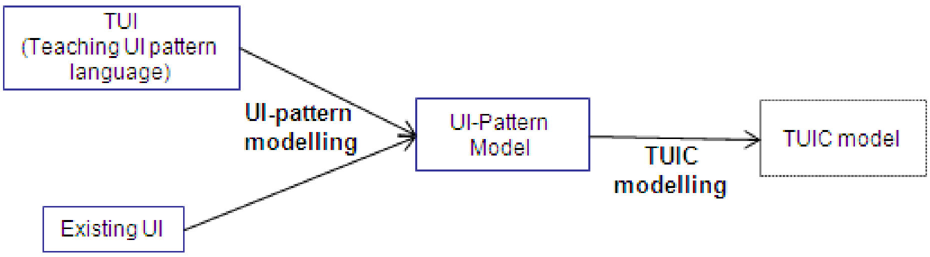


Figure 6.3 - Building pattern-based models of an existing UI

For Study One and Study Two students are introduced to using UI patterns and pattern languages by applying the modified pattern-driven UI design method as shown in Figure 6.3. As explained in Chapter Five, the original method was modified by replacing the user requirements documentation with an existing user interface.

All three studies investigate different aspects of pattern-driven UI design: how the participants go about developing their models, the attributes of the models they create and their thoughts about the method, tools and tasks. TUIC modelling is investigated in Study Two and again in Study Three.

6.7 Experimental Artefacts

A similar set of artefacts is used for each of Studies One, Two and Three although often in different versions. The artefacts are:

- The set of nineteen UI patterns that link together into the pattern language TUI,

- The UI-pattern modelling exercise,
- The Patterns Questionnaire,
- The Exit Questionnaire,
- The TUIPL framework – UI-pattern modelling method,
- The tutorial.

6.7.1 TUI Patterns

Three versions of TUI are used, each using a different pattern form. An example version of the pattern language can be found in Appendix A8. In Study One, two versions of TUI are compared, the narrative pattern form and the illustrated pattern form.

01 - Narrative
02 - High-density Information
03 - State Information
04 - Transaction by Form
05 - Control Panel
06 - Collection beside Content
07 - Master with Details
08 - Cascading Collections
09 - Groups with Titles
10 - Hierarchical Set
11 - Tabular Set
12 - Stack of Working Surfaces
13 - Choice from a Small Set
14 - Choice from a Large Set
15 - Convenient Environment Actions
16 - Pointer Shows Affordance
17 - Forgiving format
18 - Input Prompt
19 - Good Defaults

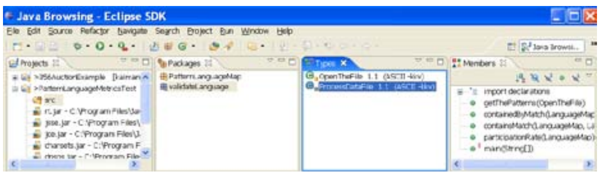
Table 6.4 - Names and numbering of patterns in TUI as used in Study One

The full set of nineteen patterns as named and numbered for Study One are listed in Table 6.4.

<table>
<tr><th>NARRATIVE</th><th>ILLUSTRATED</th></tr>
<tr><td>8A - Cascading Collections

Initial Example
Eclipse showing the Java Browsing view: projects>packages>types>members

Context
The content to be accessed is inherently hierarchical and multi-layered ….</td><td>8B - Cascading Collections

Initial Example


Eclipse showing the Java Browsing view: projects>packages>types>members

Context
The content to be accessed is inherently hierarchical and multi-layered …</td></tr>
</table>

Figure 6.4 - Sections of the two pattern forms used in Study One

Figure 6.4 shows examples of two pattern forms. The 'Initial Example' section illustrates the main difference. The narrative content of the matching patterns is the same in both versions.

Front

8 - Cascading Collections

Context

The content to be accessed is inherently hierarchical and multi-layered …

Problem

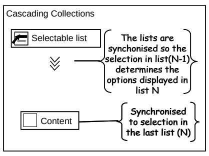
How should … locate?

Forces

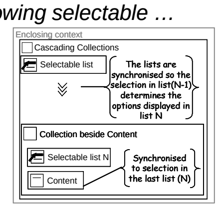
….

Solution

Express a hierarchy by showing selectable …



or



Reference

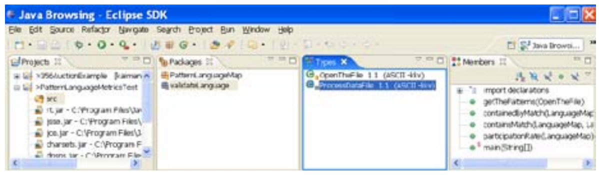
This pattern effectively …

Back

Discussion

…

Examples:



Eclipse showing the Java Browsing view: projects>packages>types>members

…

Figure 6.5 - Sections of the diagrammed pattern form used in Studies Two and Three

For Studies Two and Three the diagrammed pattern form is used. These patterns, as shown in Figure 6.5 include a TUIC model diagram identifying the essential elements of the solution. Where appropriate alternative diagrams are provided; one diagram is created using just generic symbols while the other uses active materials and may include diagrams from an associated pattern. Based on the advice of an experienced HCI educator only a subset of the CAP components and navigation components are combined to form the TUIC symbol set which are used in these diagrams (Appendix A14).

Study Two	Study Three
01 - Narrative	01 - Narrative
02 - High-density Information	02 - High-density Information
03 – Current Properties	03 – Current Properties
04 - Information on Form	04 - Information on Form
05 - Control Panel	05 - Control Panel
06 - Collection beside Content	*06 - Master with Details*
07 - Master with Details	*07 - Collection beside Content*
08 - Cascading Collections	08 - Cascading Collections
09 - Groups with Titles	09 - Hierarchical Set
10 - Hierarchical Set	10 - Stack of Working Surfaces
11 - Tabular Set	*11 -* Choice from a Small Set
12 - Stack of Working Surfaces	12 - Choice from a Large Set
13 - Choice from a Small Set	*13 - Tabular Set*
14 - Choice from a Large Set	*14 -* Convenient Environment Actions
15 - Convenient Environment Actions	*15 -* Pointer Shows Affordance
16 - Pointer Shows Affordance	*16 - Groups with Titles*
17 - Forging format	17 - Forgiving Text Entry
18 - Input Prompt	18 – Ask for Input
19 - Good Defaults	19 – Good Defaults

Table 6.5 - Names and numbering of patterns in TUI as used in Studies Two and Three

Table 6.5 lists the modified naming and numbering of the patterns used for Studies Two and Three. Two patterns are renamed for Study Two but the numbering is retained so that comparisons between images are easier. The pattern numbering is reorganised for Study Three so that the numbers better indicated the position of each pattern within the structure of the pattern language. Changes have been italicised. Pattern content has been updated to reflect changes.

6.7.2 The Modelling Exercises

Two types of model are developed by the participants. In Study One students create UI-patterns models. In Studies Two and Three, participants create a UI-pattern model and a TUIC model (Appendices A6 & A7).

6.7.2.1 UI-pattern Model

The UI-pattern-model was introduced in Section 4.2.1. It is a strongly hierarchical graph linking the patterns that describe a user interface.

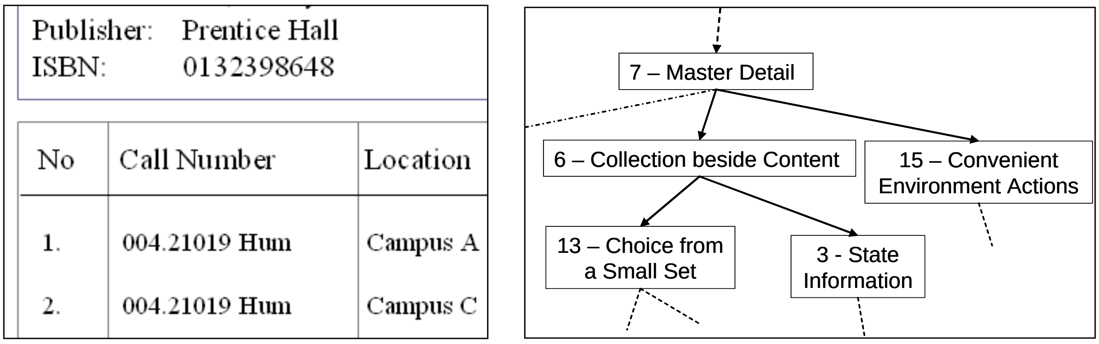


Figure 6.6 - Part of a UI with a section of the corresponding UI-pattern-model used for Study Two

The selected patterns are equivalent to the sequence of ticked patterns on a list of pattern names created using the Alexandrian method (Section 2.1.3). This sub-list of patterns can be drawn as a graph, creating the UI-pattern model. This is a hierarchical diagram based on the context and reference links between the patterns. Figure 6.6 contains a section of a simple library catalogue UI showing part of a simple master-detail form with each copy of a book displayed in a table. A section of the corresponding UI-pattern model with pattern ‘07 Master with Details’ as the context for pattern ’06 Collection beside Content’, which in turn is the context for pattern ’13 Choice from a Small Set’. The lines represent either context or reference relationships to other patterns and link the pattern nodes together. Although not shown in this example, it is possible for links to go across the levels in the hierarchy or occasionally traverse more than one level. A pattern node may have more than one link into it, especially at the lower levels in the graph and duplicate pattern nodes may be used.

6.7.2.2 TUIC model

The TUIC model diagrams associated with the solution part of a pattern can guide the development of a TUIC model based on the associated UI-pattern-model.

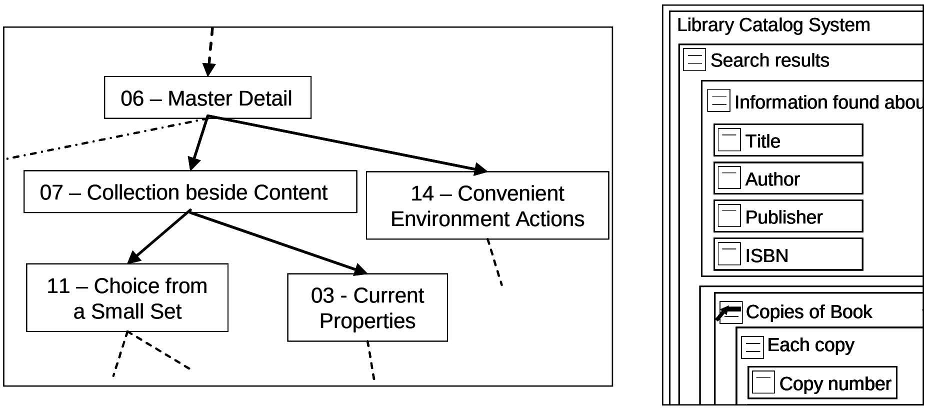


Figure 6.7 - Part of a UI-pattern model with a section of the corresponding TUIC model as used in Studies Two and Three

Figure 6.7 shows a section of the library catalogue UI-pattern model with part of the corresponding TUIC model along side. The nesting evident in the TUIC model corresponds to the levels in the UI-pattern model graph.

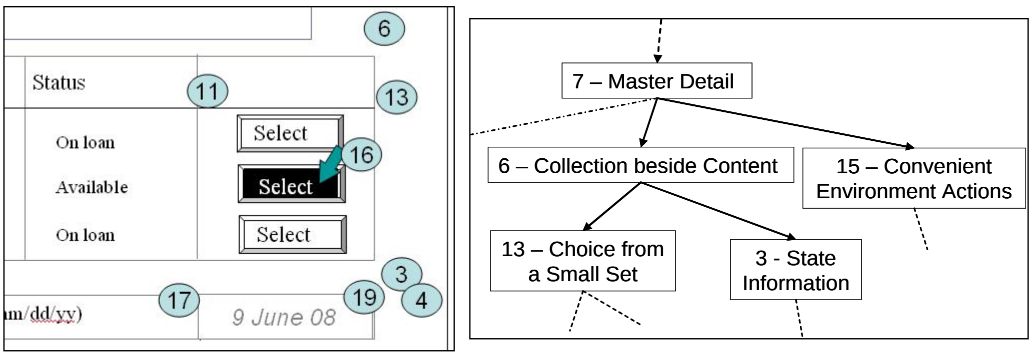


Figure 6.8 - Sections of the annotated UI and UI-pattern model.

For Exercise Two in Study Three, in addition to the UI-pattern model the participants were presented with an annotated version of the UI (Figure 6.8). The annotations indicate which patterns describe that section or component of the UI. This is an alternative way of representing the UI-pattern model.

6.7.2.3 The Workbook

The material and example used to introduce UI-pattern modelling and TUIC modelling from Studies Two and Three are used as the basis for creating a self-teaching workbook. The illustrations are similar to those used in PowerPoint slides for the introduction to the tutorials for studies One and Two. The workbook is designed to introduce how to create the models for a new user interface based on a set of user requirements. The complete workbook can be found in Appendix A5.

Associated with the workbook is a set of exercise artefacts. These include a set of requirements for a new user interface for the first exercise, and a model UI-pattern model exemplar for the second exercise. The example is different to that used in the tutorials for Studies One and Two so that it will be possible to use this exercise in a subsequent class tutorial.

6.7.2.4 Exercise Instructions

Each exercise in Studies One, Two and Three has an associated instruction sheet which includes the steps in the method to be used and a worksheet on which to draw the resultant model. These can be found in Appendix A6. The instructions for Exercise One in Study Three are accompanied by a set of UI requirements. Studies One and Two have an image of the UI to be modelled. The instructions for Exercise Two are accompanied by an exemplar UI-pattern model.

6.7.3 The TUIC Test

The TUIC components test is administered after the TUIC modelling exercise in the second part of Study Two only. This is a post-test to evaluate what the students have learned about the basics of the TUIC notation and is limited to just the CAP symbols. It contains only five questions with twelve evaluation points. The test contains three multi-choice questions, one naming question and one matching question. The test can be found in Appendix A13.

6.7.4 The Patterns Questionnaire

The Patterns Questionnaire is administered after the UI-pattern modelling exercise. The questionnaire is based on the de facto industry-standard software usability questionnaire SUMI: Software Usability Measurement Inventory (Kirakowski 1994). Amending an existing survey instrument is preferable to starting from scratch (Kitchenham & Pfleeger 2002a). The aim of the Patterns Questionnaire is to assess the use of the patterns. The questionnaire's statements have been adapted to suit UI patterns and cover issues such as: how the students feel about using the UI patterns, the ease of learning their content, helpfulness and ease of use.

The statements making up the different versions of the Patterns Questionnaire cover six topics:

- Building the UI-pattern model (3 statements),
- Aiding discussion (2 statements),
- Pattern language structure (4 statements),
- Information content of patterns (6 statements, 1 additional for studies Two & Three),
- Illustrations (3 statements for studies Two & Three),
- Pattern names (2 statements for studies Two & Three).

The Patterns Questionnaire used in Study One is modified for Studies Two and Three (Appendix A11) to include additional statements taking into account the TUIC model diagrams and adding statements covering pattern naming. Twenty-one statements are common to Studies One, Two and Three. Another five statements are common to Studies One and Two. These five statements are replaced in Study Three with statements more appropriate to the design process being applied. Study One has four additional statements and Studies Two and Three have six additional statements. Each statement in the questionnaire requires students to indicate whether they agree, disagree or are undecided.

6.7.5 The Exit Questionnaire

The Exit Questionnaire is administered at the end of each session for Studies One, Two and Three. A copy of the different sections can be found in Appendix A12. This questionnaire gathers information on the respondents' opinions and impressions of the exercises they have completed. It also provides an opportunity for the participants to suggest improvements. Most of the five sections include one question that requires participants to rate the topic on a five point scale from unhelpful to extremely helpful. This scale is reasonably balanced with the end points being opposites and the intervals between being evenly distributed (Kitchenham & Pfleeger 2002a). No neutral option is provided to ensure respondents express either a positive or a negative opinion.

Section A gathers background information about demographic, education and UI development experience. Study Three is a series of case studies investigating the development of a new UI by professional developers therefore the type of information required is different to that required from the student participants of the preceding studies.

Section B asks participants to consider pattern layout and content and is used for all three Studies. Study Three is administered in two sessions. Two versions of the questions are used: one after completing the UI-pattern model and one after completing the TUIC model. One question has been modified to refer to each of the appropriate modelling methods.

Section C on UI-pattern modelling and **Section D** on TUIC modelling are very similar. Both versions include steps specific to the method used. The first group of questions asks participants to suggest modifications and the second asks about possible omissions. Then participants are asked for any other suggestions for improving the method. The last question has two versions. For UI-pattern modelling, the question asks the participants to rate the modelling technique. For the TUIC modelling, the question asks participants to rate how useful patterns are in helping them learn about the technique.

Section E is a short section which asks the student participants to report on the use of patterns as an aid for helping them communicate with their partners. This section is part of the questionnaires for Studies One and Two because the students are working in pairs.

Section F is used only for Study Three and asks the participants' opinions on aspects of how UI-pattern modelling might aid non-professional members of a design team communicate their UI needs and ideas to the rest of the team. **Section G** is similar but the questions ask for opinions associated with TUIC modelling.

6.8 Data Analysis

For each study the results from data collected using more than one technique are triangulated (Wilson 2006, Benbasat *et al.* 1987, Yin 1994). Confirming results from two or more sources reduces threats to validity but does not eliminate them. Sets of data and analysis details can be found in Appendices A15, A16, A20 & A23.

Each of the data collection methods used in each of the studies has a set of objectives relating to objectives of the associated study. They are introduced and discussed in the appropriate sections reporting on the results of each study.

6.8.1 Observation

The approach to analysing the observations for Studies One and Two is similar to a multiple case study approach (Galliers & Land 1987) with each pair of students being a mini case study. The data is collected by analysing the sequences of images of each case study using explanatory narrative (Gerring 2004, Flyvbjerg 2006). When actions such as pointing and focussing on specific items occurred often enough, the counts were recorded. Sequences of actions are identified as they may establish whether the students follow the proposed method. Time to complete building the different models is established as improved time-to-complete a task can be used as surrogate indicator of an improved or successful process (Lin & Landay 2008, Wania & Atwood 2009).

In studies One and Two observations by the researcher which appear to be relevant are recorded when possible. These are used only as confirmation of other findings. In Study Three no images will be taken and the researcher is in the role of a participant-observer (Gerring 2004, Kaplan & Duchon 1988, Yin 1994).

6.8.2 Models

For Studies One and Two, the results from analysing the scores of the models are indicative only because the validity maybe compromised by students submitting their own version of each model although they are encouraged to consult with their partner. Where there is enough variation in the scores the models are analysed individually.

Scoring criteria are defined in Appendix A15. Four criteria (Appropriate patterns, Potential patterns, Appropriate links and Potential links) are used to score the UI-pattern models. Four criteria are used to score the TUIC models (Appropriate TUIC symbols, Potential TUIC models, Correct Nesting and Appropriate Labels). Scores are reported using descriptive statistics (minimum, maximum, average and standard deviation) for presenting results from Studies One and Two.

Student’s T-test is used where scores of the models need to be compared because it is a robust test for analysing small sample sizes. In Study One scores for UI-pattern models created using the different pattern forms are compared. Also the scores from Exercise One and Exercise Two are compared to determine whether learning has taken place.

It is difficult to design tasks that directly measure attributes of methods such as goodness or effectiveness, but it is possible to use surrogate measures (Pfleeger 1994, Kitchenham 1998). Model scores are used as a surrogate to indicate the successful application of the associated modelling method. These results can be compared to those from self-reporting on the usefulness of the method and observations of student involvement in the process.

In Study One the versions of TUI have a structure similar to that of other UI pattern languages (Section 5.4). The distribution of the different link types connecting the patterns used to create the UI-pattern models can be determined as can the distribution of link types occurring in TUI (Figure 5.6). To investigating whether students pay attention to the link structure of the pattern language the two distributions can be compared using Chi-square. This comparison is indicative only, firstly because the sample is small and secondly because the UI-pattern models only employ a subset of TUI.

6.8.3 The TUIC Test

The results from scoring the TUIC test are presented as simple descriptive statistics. The results of this test are only indicative but they can be correlated with the scores from the TUIC models to provide additional evidence as to whether the students successfully learned about the TUIC modelling.

6.8.4 Patterns Questionnaire

For all studies, analysis of the data from the Patterns Questionnaires should help ascertain participants' opinions of their experience with using the different versions of TUI. The data is coded and then ranked. The responses to each statement are coded as '-1' for disagree, '0' for undecided and '1' for agree and totals calculated. For studies One and Two the mean provides an overview of students' perception of using patterns. Simple counts and totals are used to represent sub-groups of questions associated with different topics such as method or illustrations.

For Study One, before comparing students' perceptions of using the different pattern forms, totals are correlated to check whether students have answered independently because students worked in pairs.

Because this questionnaire employs a three point scale Chi-square is used to compare responses associated with each pattern form. The matrix contains pattern form and responses. Both Likelihood Ratio and Pearson Chi-square values to two degrees of freedom are calculated. When results are similar only Pearson Chi-squared is reported. A similar approach is taken to test difference in response to each sub-group of questions.

6.8.5 Exit Questionnaire

The exit questionnaire is mostly open ended questions which are reported descriptively. For the questions that required a response on a five point scale from 'very helpful' to 'unhelpful' percentages are reported. Counts with an associated bar graph and some percentages are sufficient to illustrate these responses.

6.9 Procedures Followed

Each of the Studies follows a similar process. For Studies One and Two the experiments are part of an existing third year university HCI course. Students taking part attend an introductory lecture prior to the tutorial. The time available for the tutorial is nominally two hours. These tutorials are structured as problem based learning (PBL) exercises although time constraints limit them.

As reported in Chapter Four, Merrill (2002) identifies five principles of PBL which include the constructivist view of learning: the problem, the activation phase, the demonstration phase, the application phase and the integration phase. Re-engineering an existing UI is a valid design problem regularly untaken by professional UI designers and therefore represents an authentic real-world problem.

The tutorials follow the four phases of effective PBL instruction. They are embedded in an existing UI course of study and are placed near the end of the program. Other than the introduction of UI patterns and TUIC modelling all the UI theory and practice knowledge contained in the patterns is effectively revising material already taught, although packaged and presented differently. This ensures that the students build new knowledge on a foundation of existing knowledge.

In the demonstration phase an example UI-pattern model, or TUIC model as required, is created. This demonstrates both the application of the TUIPL framework and the process of building the model itself. During the application phase, students work with their partner applying the appropriate modelling method for the UI design phase to complete their own UI-pattern model or TUIC model. The exemplar model is available for students to reference at anytime during the tutorial and the researcher is available to scaffold student learning when help is requested. During the modelling activity each pair of students are encouraged to discuss pattern content, selection choices and collaborate on building their models.

Time constraints preclude any reporting back to the class by each pair of students but as part of the research the students independently complete questionnaires that include open ended questions asking them to reflect on different aspects of the processes and tools they use during the tutorial. The modelling methods the students apply are flexible enough to allow them to develop their own forms of intermediate or transitional representation.

All the studies begin with an introduction to the research activity which includes an explanation of participants' rights and a request for the participants to agree to take part. Ethics approval forms are distributed and after completion are collected and placed in a sealed envelope. The introduction includes a brief overview of UI-patterns, reviewing information covered in the earlier lecture. Building an exemplar UI-pattern is demonstrated and made available for reference throughout the exercise. After completing the UI-pattern modelling exercise the participants complete a Patterns Questionnaire.

The introduction to TUIC modelling also includes the demonstration of a worked exemplar that is made available for examination throughout the prototyping exercise. After completing the TUIC model, the CAP test is completed individually. At the end of the tutorial session an appropriate version of the Exit Questionnaire is administered.

For Studies One and Two the models created by the students are copied and the originals forwarded to teaching staff for assessment. The teaching staff for the HCI course use independent assessment criteria to evaluate these models.

In teaching students architectural design, working with patterns was shown by Davis (1986) to encourage students' communication about the problem they are solving. He says:

> *"Asking the students to define their design intentions in the form of such patterns promotes discussion that helps lead to the needed consensus." (ibid, p15)*

Encouraging students to collaborate is especially important in the constructive learning approach (Merrill 2002) and discussion between the students working towards a common goal is central to the collaborative learning process. Therefore the students are encouraged to work cooperatively in pairs, i.e. collaborating on their UI design modelling but with all assessment being individual (Prince 2004).

6.9.1 Preparation

For Studies One and Two, the room is re-organised before the tutorial to maximise space between pairs of students. This is to limit communication between pairs and allows room for the photographer to move easily from pair to pair. The information sheets and Ethics approval forms are laid out and pencil and erasers made available. A message on the board asks students to select a partner (within their group for Study One) and sit at one of the prepared places.

At the start of the tutorial the lecturer introduces the researcher then leaves the room.

All the artefacts are packaged so that they can be easily handed out. All the material for each exercise is in the package even though this requires some degree of duplication. All material is collected between exercises and the next package distributed.

6.9.2 Study One: Presentation Procedure

The educational aims of the Study One tutorial are to introduce students to UI pattern languages and help them:

- Become familiar with the organisation and content of UI patterns,
- Acquire an understanding of the structure of a UI pattern language,
- Create a user interface pattern-model (UI-pattern-model) describing an existing user interface.

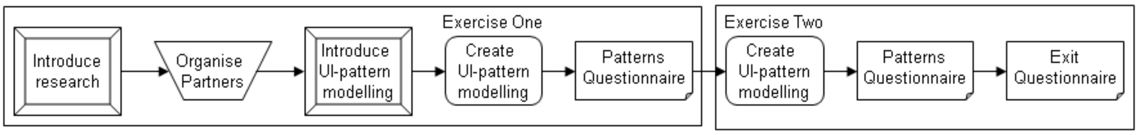


Figure 6.9 - Tutorial procedure for Study One

The procedure for the tutorial is shown in Figure 6.9. Before the tutorial the students are allocated to one of two groups. Students chose a partner to work with from within their allocated group. The exemplar in the introductory tutorial is for a small building using the Alexandrian pattern language. The appropriate set of patterns, the example UI, exercise instructions and the exercise worksheets are distributed. The students work with their partner to complete a UI-pattern model representing the example UI. After completing and submitting their model the students complete a Patterns Questionnaire and then have a short break before starting the second exercise.

The second exercise is similar to the first but for a different UI example and using a different set of patterns. After completing the Patterns Questionnaire associated with this exercise, the students complete the Exit Questionnaire.

6.9.3 Study Two: Presentation Procedure

The educational aims of the Study Two tutorial are to introduce students to UI pattern languages and help them:

- Become familiar with the organisation and content of UI patterns,
- Acquire an understanding of the structure of a UI pattern language,
- Create a user interface pattern model (UI-pattern-model) describing an existing user interface,
- Create a user interface conceptual model (TUIC model) described by the UI-pattern model.

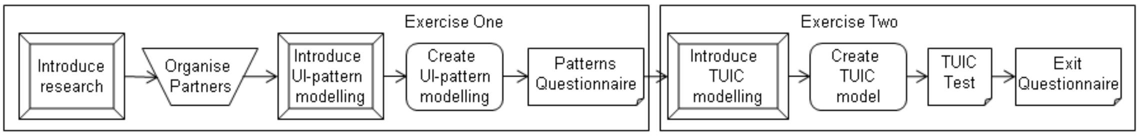


Figure 6.10 - Tutorial procedure for Study Two

Figure 6.10 shows the procedure for this tutorial. Exercise One in this study is similar to Exercise One in Study One, but applies the modifications based on the findings from Study One. Exercise Two takes a similar approach starting with an introduction to TUIC modelling. After completing the test students complete a version of the Exit Questionnaire that includes an additional section covering TUIC modelling.

6.9.4 Study Three: Case Study Procedure

This study does not involve students so there are no educational aims. The procedure is similar to that of the previous two studies but rather than a lecture style introduction the researcher does a one-to-one demonstration using the UI exemplar and the workbook.

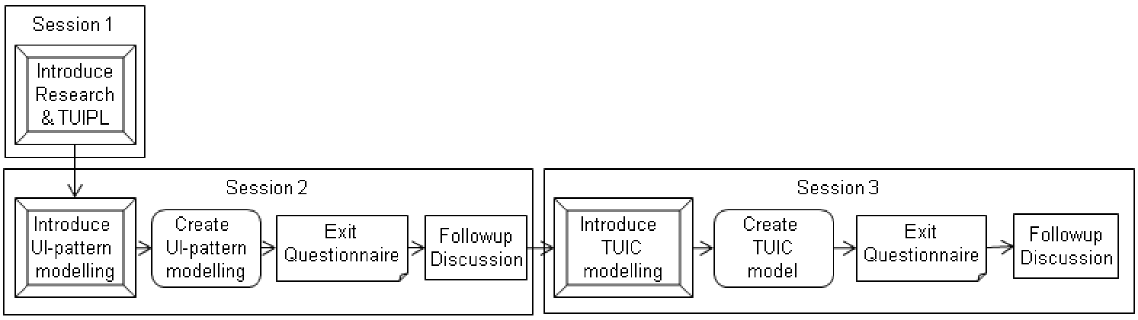


Figure 6.11 - Case study procedure for Study Three

The study is split into three sessions with each participant as shown in Figure 6.11. The introductory session is short and includes an explanation of the research, the ethics approval process and the completion of the first section of the "Exit Questionnaire" to collect background information. The participant is introduced to the pattern set and given a copy to study at their leisure.

Each of the two main sessions occurs between five to ten days apart. Session Two starts with familiarisation with the UI-pattern modelling method and the requirements of the interface to be designed. Then the participant creates a UI-pattern model matching the requirements. On completion of the modelling exercise they complete the relevant sections of the Exit Questionnaire. At the end of the session the researcher discusses observed behaviours to clarify any interpretations and asks for any additional feedback from the participant about the exercise and the artefacts.

In Session Three the participant is first introduced to TUIC modelling then re-acquaints themselves with the UI requirements and an exemplar UI-pattern model for the new user interface. They use this UI-pattern model to guide the creation of the TUIC model. On completion of the modelling exercise they fill out relevant versions of the questionnaires. A final discussion clarifies observations and allows for any additional comments from the participant.

6.10 Summary

As noted in Chapter Five, to be able to scaffold students' learning the teacher needs to understand which activities and representations will help the students understand the problem and enable them to solve it (Jonassen 1999). This chapter identifies the Design Study as the method providing the framework for this research. It is exploratory in nature and multiple research methods are used: the quasi experiment, the pre-experiment and the case study. Study One investigates two UI pattern forms. Studies One and Two aim to establish whether students, with the help of TUIPL, can develop conceptual models of existing user interfaces in the form of UI-pattern models and TUIC models. Study Three aims to establish primarily that by following TUIPL students are provided with an authentic experience and to check that the methods can be used to develop UI conceptual models for a new user interface.

Eight data collection methods are identified. Five of these are for use in all three studies. Digital images taken at regular intervals are for use in Studies One and Two. The case studies of Study Three include an informal discussion at the end of each session. Multiple sources of data are required so that by triangulation potential conclusions can be verified from more than one source (Benbasat *et al.* 1987, Yin 1994, Mingers 2001, Kauloi 1998).

The processes in TUIPL to be tested by Studies One and Two are identified. The artefacts required to carry out the studies are introduced. The details of the different versions of TUI are presented. They are referred to respectively as: the ‘narrative’, ‘illustrated’ or ‘diagrammed’ versions. The content and layout of the pattern forms are the same but the illustrated form includes an example that is associated with a snapshot of an example screen. The diagrammed form includes an additional TUIC model diagram illustrating the essential elements of the solution. Examples of the UI-pattern model and the TUIC model are presented and the relationships between these two models established.

The details of two types of evaluation questionnaire and the TUIC test are described. The Patterns Questionnaire is a combination of six different groups of statements. The Exit Questionnaire is a combination of seven sections covering: background information, UI patterns, UI-pattern models, TUIC models and the associated modelling methods, communication and designing new UIs. The questions mostly consisted of open-ended questions.

The processes for presenting the tutorials in Studies One and Two are described as is the approach used for the case studies making up the third study. Details of each of the three studies are presented in the following four chapters.

Chapter 7: Study One - Developing UI-pattern Models

This first experiment investigates student reactions to using patterns to develop UI-pattern models applying the UI-pattern modelling method from TUIPL and comparing the use of two versions of TUI. It focuses on elucidating student experiences and views in order to:

Explore the use of a user interface pattern language in the creation of UI-pattern models by student UI developers.

The objectives for this study are to:

1. Determine whether UI patterns are an acceptable medium for presenting UI information to students.
2. Investigate whether illustrated examples influence student understanding of patterns and pattern languages.
3. Determine whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model.
4. Discover whether building a UI-pattern model develops student understanding of UI patterns and pattern language structure.

First the experimental design and student demographic data is presented, then the results from each of the different data sets are described. These results are consolidated in the discussion section and conclusions are reported with recommended improvements to tutorial artefacts and the UI-pattern modelling method to enhance student learning.

7.1 Research Protocol

Study One investigates primarily the second of the research questions and in part, the third research question (Section 1.3). The Study compares students' use of the first two versions of TUI (Section 6.7.1). The students are introduced to UI-pattern modelling via TUIPL (Section 6.7.2.1).

Each data collection method (Section 6.3: observation, models, pattern questionnaires, exit questionnaire) used in this study also has specific objectives related to each of the three objectives of this study. These are introduced and discussed in the appropriate sections.

7.1.1 Experimental Design

A within-subjects design was used for this experiment. The subjects are undergraduate students enrolled in an HCI course (Section 6.5). Before the tutorial the students were paired, based on the results of a prior UI exercise, then allocated randomly to either Group A or Group B to ensure groups of similar ability.

Patterns	Illustrated	Narrative
Exercise1	A	B
Exercise2	B	A

Table 7.1 - Pattern of treatments of the two groups of students A & B

The research used two treatments as shown in Table 7.1. Each treatment used a different pattern form: narrative and illustrated. Because both treatments are applied to both groups of students, a different UI example is used in each exercise. Using different pattern forms and different UI examples should minimise the carryover learning of pattern content between exercises. Group A used the illustrated patterns for Exercise One and the narrative patterns for Exercise Two, with the reverse for Group B.

To replicate a teaching situation, students chose a partner from within their assigned group. Students worked with their partner and could ask the researcher questions about patterns (although not about solutions). The three pairs of students in Group A are referred to as Cases 1, 2 and 3. The four pairs of students in Group B are identified as Cases 4, 5, 6 and 7

Time constraints were a limiting factor in the tutorial so activities were selected to maximise outcomes. No pre-test was administered because no student had any experience with UI-pattern modelling. The students were in the final week of a semester-long HCI course. It was unlikely that results of a pre- and post-test design would be useful because there was no additional information contained in the patterns.

The procedures and artefacts were trialled using two pilot studies reported in Appendix A3. The data analysis methods were introduced and discussed in Section 6.8.

7.1.2 Activities and Artefacts

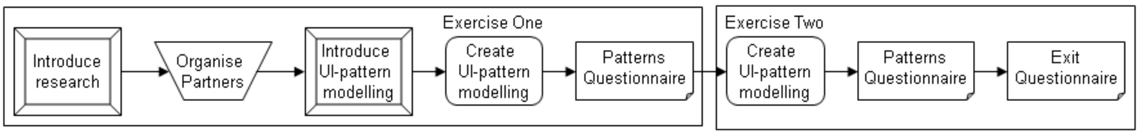


Figure 7.1 - Tutorial procedure for Study One (Copy Figure 6.9)

Figure 6.9 provides an overview of the tutorial procedure used and the list of the tasks for each of the exercises. The tutorial starts with students reading the information sheet about the research and completing ethics approval forms. These are collected by the assistant so the researcher has no knowledge of who agreed to take part in the research.

The students were provided with an instruction sheet and an example UI to model (Appendix A6, Section A6.3). A worksheet was provided on which the resulting UI-pattern models were to be submitted (Appendix A7).

Ethics approval documentation: * **Information Sheet** * **Consent Form**
For each of the UI-pattern modelling exercise: * A version of TUI * A UI example * Instruction sheet outlining the procedures * An answer worksheet for the UI-pattern model * A Patterns Questionnaire
The exit questionnaire

Table 7.2 - Artefacts the students used in Study One

Table 7.2 lists the artefacts the students used. The Patterns Questionnaire (Appendix A11) was administered after each exercise and was primarily concerned with collecting student's opinions of the patterns they had just used. The sections of the Exit Questionnaire (Appendix A12) these students completed included:

Section A to collect back ground information

Section B to collect data about the patterns

Section C to ask the students to reflect on the method followed to build the UI-pattern model

Section E to ask about communication with their partner

7.2 Overview of participants

All fourteen students enrolled in a first HCI course agreed to participate in this experiment.

DEGREE PROGRAMME	NUMBER
Bachelor of Arts	1
Bachelor of Business	1
Bachelor of Business Information Systems	1
Bachelor of Engineering	4
Bachelor of Information Systems	2
Bachelor of Science	5

n= 14

Table 7.3 - Degree Programs students are completing.

Table 7.3 shows the wide range of degree programs the students are enrolled in. Demographic data for students in each case study is not provided as the sample size is too small and individuals may be identifiable from such a description. They ranged in age from twenty to thirty years. Only three were female. Only two students reported some IT industry experience involving the development of user interfaces. Their self-evaluation of proficiency in designing a new user interface ranged from experienced to low. Three students qualified their response with a comment indicating that they were average, based on experience gained during the HCI course. By coding their responses on a scale of one to five, four students reported they are above average, four below average and six considered themselves to be average.

7.3 Results from Exercises One and Two

The results of analysing the data are summarised in the following sections.

7.3.1 Observed Behaviours

The objectives for the observations were to:

1. Identify ways the students manipulate the artefacts.
2. Identify sections of the patterns that students focus on.
3. Ascertain whether the students are following the UI-pattern modelling method.
4. Identify whether there are differences in how the students use the two pattern forms.

The details of the observations are described in seven case studies in Appendix A17. The observation criteria that guided the analysis of the images is in Appendix A10.

	Cases	
Transformation Model	**Illustrated**	**Narrative**
Hierarchy building		1, 6, 7
List marking		4, 5, 6, 7
Rough copy	2, 7	4, 5

Table 7.4 - Students observed creating intermediate transformation models

Table 7.4 shows the three forms of transitional representation identified. All were first observed in Exercise One, and mostly when working with the narrative form of the patterns. Some of the seven cases used more than one of these activities.

Three instances of building pattern hierarchies by manipulating the narrative form of the patterns into the required structure were observed, for example as shown in Figure 7.2.

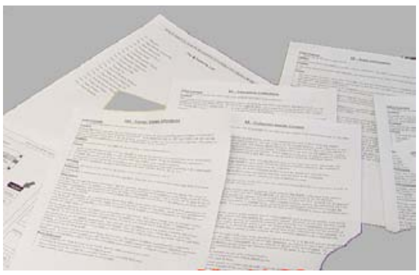


Figure 7.2 - Building a pattern hierarchy

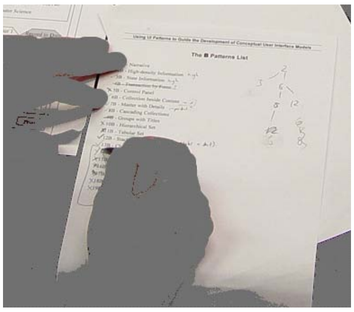


Figure 7.3 - Working directly on pattern list

Four instances of marking the pattern list were observed (Figure 7.3). Four cases of creating a rough copy were also observed. Two instances occurred when students worked with the illustrated patterns and two with the narrative form.

Focus of pointing	Count
Instructions	1
UI component	8
Names on pattern list	2
Patterns	27

Table 7.5 – Pointing to artefacts counts

As can be seen in Table 7.5 most of the observed pointing behaviour included patterns, as would be expected, because patterns were central to the UI-pattern modelling exercise.

	Narrative	Illustrated	Counts
Name			
Initial illustration		20%	2
Context section	6%	10%	2
Problem section	6%		1
Solution section	18%	10%	4
Forces section	18%		3
Reference section	41%	40%	11
Discussion section	12%	20%	4
Examples			
% of images	35%	24%	

Table 7.6 - Percentages of pattern sections pointed partitioned by pattern form with total counts

The breakdown of pattern pointing behaviour is shown in Table 7.6 Most pointing is associated with the reference section indicating students predominantly followed the top-down approach of the method.

Items in Focus	Narrative	Illustrated	Counts
UI - list	2%	3%	2
UI - pattern	25%	53%	31
UI - graph	2%		1
graph - list	5%		2
graph - pattern	2%	3%	2
list - pattern	7%	11%	7
pattern - pattern	57%	32%	37
% of images analysed	92%	90%	

Table 7.7- Percentage of items in focus partitioned by pattern form with total counts

Table 7.7 shows data describing items students were focussed on. The ratios of UI-with-pattern to pattern-with-pattern focus items (greyed lines) indicate that when using the narrative patterns the students are more focussed on matching pattern content.

The analysis of behaviour in the image sequences indicates that most of the students carried out the activities identified in the method in the prescribed order.

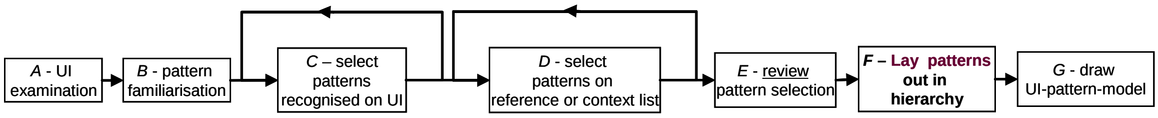


Figure 7.4 - Apparent method used by students in Case 1 including hierarchy building

The students followed the method because the basic structure underlies all their modifications (Figure 5.4). Most of the students elaborated the method by adding additional activities (e.g. Figure 7.4). These included: creating an intermediate transformation model using hierarchy building, marking the list, creating a rough copy and reviewing models at some stage before submitting them.

Time taken to complete each exercise ranged from twenty-two minutes to thirty-five minutes (Appendix A16), but to complete the actual modelling activity the range was from twenty-three to twenty-eight minutes. A closer examination of the image sequences showed that when working with the illustrated patterns on Exercise One, all students had completed copies of the UI-pattern model within twenty-five minutes, although one was a rough copy. For most of the cases (4, 6 & 7) working with the narrative patterns it is clear that after twenty-five minutes had elapsed none of these students were even at the stage of having a partially completed UI-pattern model.

They were still struggling with the concept of creating a linked graph. The most graphic example is Case Four where after twenty-eight minutes the students have a graph with pattern nodes linked by many edges and a large cross drawn through it (Figure 7.5). At this point they requested help.

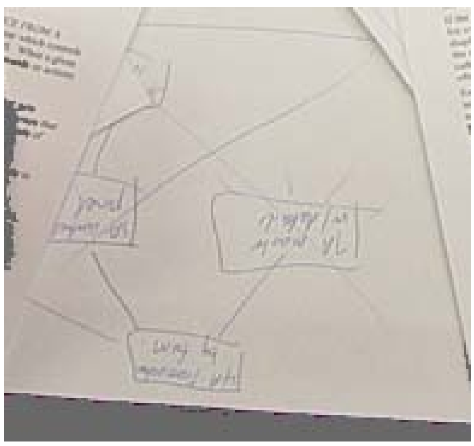


Figure 7.5 - Crossed out first attempt at UI-pattern model building

By comparison, Cases, Six and Seven at this time had partial graphs containing three and two unlinked nodes respectively

	Exercise 1	Exercise 2	Average
Illustrated	25	24	24
Narrative-only	32	23	28
Average	28	23	

Table 7.8 – Comparing time taken in minutes to build a UI-pattern model

Table 7.8 shows that students took on average twenty-eight minutes to complete the UI-pattern model in Exercise One and twenty-three minutes to complete it in Exercise Two.

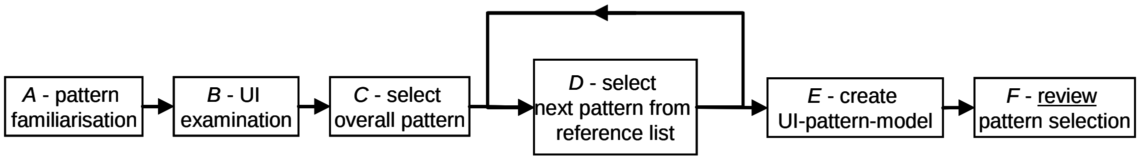


Figure 7.6 - Apparent method used by students in case 3 concluding with a review activity

Some of the students were observed monitoring their progress by reviewing pattern selections as they built their models, for example Case 3, as shown in Figure 7.6. Reviewing behaviour occurred with both pattern forms.

When students asked for clarification on how to create a UI-pattern model, referring to the architectural model used in the introduction did not appear to be particularly helpful. Many queries were associated with inconsistencies in the matching of reference and context sections of related patterns.

Students were seen to move, compare, point to patterns and also to draw on other artefacts as they built their UI-pattern models. Only one pair of students did not create a transformational representation. Most pattern sections were pointed to at some stage by at least one student, but over forty percent of pointing behaviour was directed towards the Reference section. The sequences of images associated with each case shows that the students followed the essential steps in the method with some variation.

There are distinct differences in how the students use the two pattern forms. They include the time to complete models in the first exercise, and the amount of pattern-to-pattern and UI-pattern pointing behaviour.

7.3.2 UI-pattern Model Evaluation

The objectives for scoring these models are to:

1. Identify whether there are differences between the models based on the pattern form being used.

2. Ascertain whether students have developed an understanding of pattern language structure.

The marks for the models are in Appendix A16 and the marking scheme in Appendix A15. Student's T-test was used to determine whether the scores for the models created by the students when working with the two pattern forms were different.

	Appropriate patterns		Potential patterns		Appropriate links		Potential links	
	illustrated	narrative	illustrated	narrative	illustrated	narrative	illustrated	narrative
min	71%	43%	44%	33%	50%	42%	29%	43%
max	100%	86%	56%	78%	75%	83%	43%	71%
average	84%	67%	52%	57%	58%	61%	38%	59%
std	13	18	6	18	13	18	7	14
p T<t	0.07		0.53		0.75		0.01	

Table 7.9 - Comparing marks out for the UI-pattern models from Exercise One partitioned by pattern form

Table 7.9 shows the summary statistics for Exercise One. When using the narrative pattern form the students appear to have paid more attention to the link structure of the pattern language. They identifying all potential links which was significantly better (p=0.01) than when they used the illustrated pattern form. Success in following the links when using the narrative form possibly explains the improved (but not significant) performance in finding potential patterns when using the narrative patterns. The more successful students used the reference and context sections to identify and confirm links. Attention to these sections would also identify other candidate patterns for them to consider. No significant differences between the four categories were found for Exercise Two.

When using the illustrated patterns it is clear that the students were better at eliminating inappropriate patterns because the percentage of appropriate patterns in their UI-pattern models were higher than those created when using the narrative patterns, although only weakly significant (p=0.07).

	Appropriate patterns		Potential patterns		Appropriate links		Potential links	
	Exercise1	Exercise2	Exercise1	Exercise2	Exercise1	Exercise2	Exercise1	Exercise2
min	43%	77%	33%	45%	42%	13%	29%	7%
max	100%	100%	78%	91%	83%	75%	71%	50%
average	74%	94%	55%	65%	60%	51%	50%	26%
std	18	11	14	15	16	24	16	12
p T<t	0.00		0.08		0.25		0.00	

Table 7.10 - Comparing UI-pattern model marks from the two exercises

The scores for each exercise are shown in Table 7.10. The model for Exercise Two was more complex. The students clearly did significantly better at identifying appropriate patterns (p=0.00) and apparently better at identifying all potential patterns. If learning took place during the exercises the expectation was that students would perform at least the same or better in Exercise Two, in particular improving their understanding of the link structure connecting patterns. They clearly did not, and actually performed significantly worse at identifying potential links (p=0.00).

	Improved	Same	Worse
Identifying appropriate patterns	12	2	0
Finding all potential patterns	4	7	3
Identifying appropriate links	4	3	7
Finding all potential links	2	2	10

Table 7.11 - Comparing individual student's performance for Exercise One and Two

A second method for comparing the models was to examine differences between individual student's marks for models completed during the two exercises. Table 7.11 shows that fifty percent of the students improved or performed approximately the same in adding appropriate links to their models. But, seventy-one percent were worse at finding all the potential links during Exercise Two. It does not appear that students improved their understanding of pattern language structure during Exercise Two.

Category	matched	context	reference
Pattern map for TUI	41%	32%	27%
Combined valid links in students' models	52%	25%	23%

Table 7.12 - Percentages in link categories for the TUI pattern map and student's models

Table 7.12 shows the different categories of link making up the pattern map (Figure 5.6) compared to those depicted in the students UI-pattern models. This indicates that the students were following the link information in both the reference and context sections. A Chi square result of 4.976 with a p-value of 0.083 at 2 degrees of freedom is weak support for the null hypothesis that the link category percentages of students' models do not differ significantly from those in the TUI pattern map.

For Exercise One there is a clear difference in the models the students created using the different pattern forms. The models created using the narrative version of TUI contained a higher proportion of the potential links and had fewer inappropriate patterns. There is a distinct difference in the marks and the composition of the models created in the two exercises indicating that students were becoming more familiar with the structure of the pattern language.

7.3.3 Patterns Questionnaire Results

The objectives for the patterns questionnaire are to:

1. Ascertain whether the students considered patterns helpful.

2. Identify whether the students' responses indicate differences in their perceptions of the two pattern forms.

3. Determine students' views about building the UI-pattern model.

4. Indicate whether patterns aided students' discussion about UI concepts.

5. Identify whether students developed an understanding of pattern language structure.

6. Determine students' opinions of the accessibility of the information contained in a pattern.

The Patterns Questionnaire is found in Appendix A11 and the data in Appendix A16, Section A16.3. The responses to each statement have been re coded as '-1' for disagree, '0' for undecided and '1' for agree. The totals are calculated to provide a ranking for each statement.

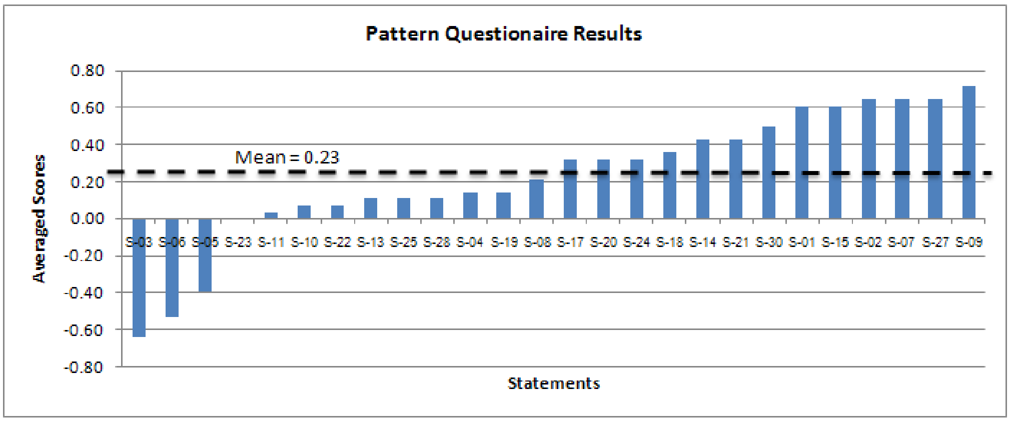


Figure 7.7 - Rank-order graph of the responses to each statement

Figure 7.7 shows the statements ranked from negative to positive. This graph indicates that the students' overall opinions of patterns are positive with an average of 0.23.

When explicitly asked their opinion on whether patterns were helpful the students responded very positively (Figure 7.8) confirming the overall ranking of all the statements (Figure 7.7).

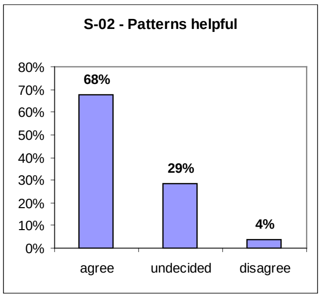


Figure 7.8 - Responses to the statement asking if patterns were helpful

Students' perception of patterns and UI-pattern model building improved between completing Exercise One and Exercise Two.

	Ex1	Ex2
agree	35.44	50
undecided	40.38	34.62
disagree	24.18	15.38

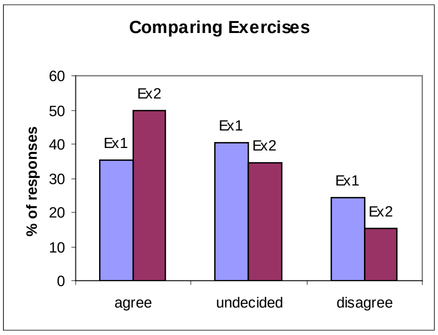


Figure 7.9 - Percentage of counts comparing responses partitioned by exercise

Their responses are shown in Figure 7.9. Accepting the null hypothesis that the students' responses would be similar for both exercises is highly unlikely with a Chi-square result of 18.742 with a p-value of 0.000 at 2 degrees of freedom.

One statement asking how straight-forward the method is (Figure 7.10), illustrates this improvement in perception. This indicates that patterns are a suitable technique for presenting user interface knowledge to students and provides further support for the first objective that patterns are helpful.

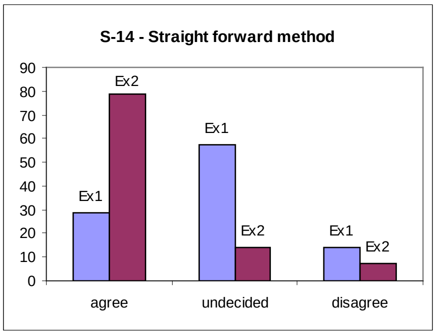


Figure 7.10 - Comparing responses to statement S-14 partitioned by exercise

Comparing responses for the two versions of TUI shows that there is a significant difference in the students' responses to using the different versions of TUI. The null hypothesis that the students' responses would be similar for both pattern forms is unlikely with a Chi-square result of 8.611 with a p-value of 0.013 at 2 degrees of freedom.

	Illustrated	Narrative
agree	47.8	37.64
undecided	35.44	39.56
disagree	16.76	22.8

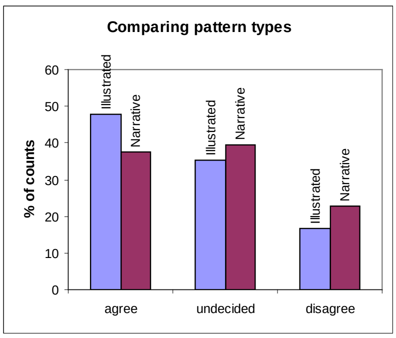


Figure 7.11 - Percentage of counts comparing responses partitioned by pattern forms

The graph and data in Figure 7.11 shows students' preference for the illustrated pattern form.

This preference for illustrations appears to have influenced students' judgement about the organisation of the information in the patterns. Although the narrative elements of the two versions of TUI were identical, Figure 7.12 shows they that perceived it to be different.

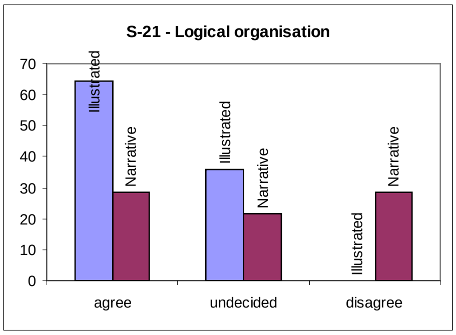


Figure 7.12 - Comparing responses to statement S-21 partitioned by pattern form

Students' opinions were sought on model building, pattern language structure, their discussions and information in the patterns, by four subgroups of statements corresponding to the last four objectives.

Topics	S-num	total
Straightforward method	S-14	12
Method easy to follow	S-28	3
Number of steps OK	S-23	0

combined n=28

Table 7.13 - Ranking totals of statements related to building the UI-pattern model

Model building: As shown in Table 7.13, although many students considered the exercises to be relatively straightforward most students were undecided about whether model building was easy.

	Ex1	Ex2
agree	17.86	32.14
undecided	26.79	41.07
disagree	55.36	26.79

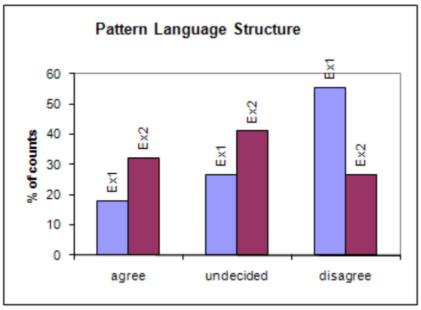


Figure 7.13 – Percentage of counts comparing responses to language structure statements partitioned by exercise

Pattern language structure: The evidence indicates that student's perceptions about using the language structure were not positive. The null hypothesis that responses to the pattern language structure statements after each exercise were similar is unlikely with a Chi-square result of 9.698 with a p-value of 0.008 at 2 degrees of freedom. But the detail of these responses (Figure 7.13) indicates that overall they were still not confident in using the structure of the pattern language.

Discussion: The graph in Figure 7.14 shows that students clearly agreed that patterns aided their discussion of both UI concepts and the exercise problems they were completing. They also helped them focus their thinking.

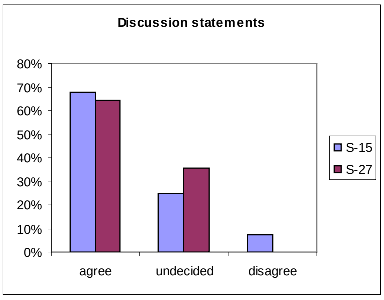


Figure 7.14 - Student responses to discussion statements.

Topic	S-num	totals
Sufficient information	S-09	20
Patterns informative	S-07	18
Clear information	S-01	17
Logical organisation	S-21	12
Patterns easy to understand	S-18	10
Content easy to remember	S-25	3

combined n=28

Table 7.14 - Ranking totals of statements related to information in patterns

Information content of patterns: The students considered the information within the patterns to be informative and accessible as shown by responses to the statements in Table 7.14. This supports the earlier finding that students considered patterns to be helpful.

7.3.4 Exit Questionnaire Results

The objectives of this questionnaire were to:

1. Determine whether the students have a preference for patterns that include illustrative images versus the narrative versions.
2. Identify those parts of patterns that students considered useful.
3. Investigate whether the UI-pattern modelling method from TUIPL for introducing UI patterns was successful.
4. Identify potential changes for improving the method.
5. Investigate whether a pattern language could provide useful guidance during UI design.

The Exit Questionnaire is found in Appendix A12 and the numeric data in Appendix 16, Section A16.4 and the comments in Appendix A18.

The students had a definite preference for the illustrated pattern form. Seventy-one percent responded positively to the relevant question. Several provided thoughtful observations about the potential problems when students place excessive reliance on illustrations. One said "*Lead to reliance on them – used them to understand the pattern not the information explaining them*".

Section	Most Useful	Least Useful
Name		1
Initial illustration	6	1
Context section	6	
Problem section	2	2
Solution section	4	1
Forces section	1	
Reference section	7	
Discussion section		6
Examples	6	3
	n=12	n=10

Table 7.15 – Sections making up a pattern

Table 7.15 shows the results of asking the students about the most and least useful sections of a pattern. The exercise of creating a UI-pattern model clearly influenced students' selections as reflected in the high scores for the usefulness of the context and reference sections. A student who identified the solution as the most useful section commented that "*It displayed the correct solution which could be visualised to help me figure out if that same solution could be applied to exercise we were given*". This student was considering the design of the example UI rather than just creating the UI-pattern-model.

Method	Count
Very helpful	6
Reasonably helpful	2
Helpful	6
Not very helpful	0
Unhelpful	0
	n=14

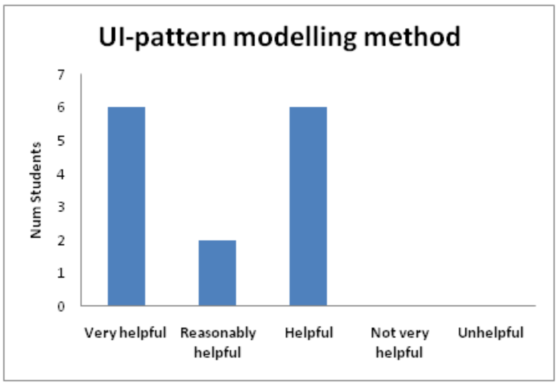


Figure 7.15 – Views of using the UI-pattern modelling method had on understanding patterns

Students responded positively when asked whether the UI-pattern modelling method was helpful (Figure 7.15). All students provided comments to support their ranking. For example one said "*It gave me an understanding of the fundamentals of patterns by actively participating in the construction of a pattern hierarchy.*" Students have suggested useful changes to consider for integration into the method for Study Two. These are to reword Step E (Figure 5.4), to include a further step asking students to record assumptions, and to modify the list of pattern names to indicate the relationships linking the patterns in the collection.

Design aid	Count
Very helpful	2
Reasonably helpful	4
Helpful	6
Not very helpful	1
Unhelpful	1

n=14

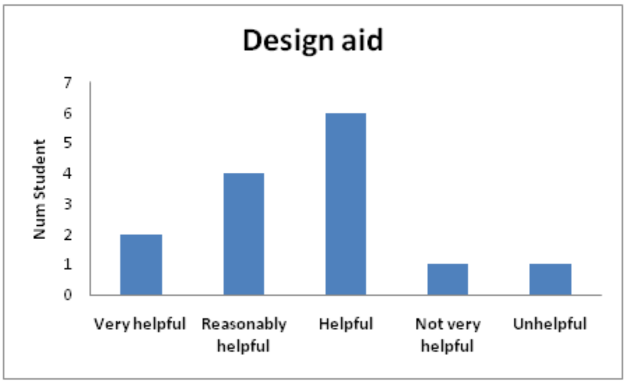


Figure 7.16 - Projected views on usefulness of patterns to aid UI design

Students considered that creating a UI-pattern model of a user interface would have a positive influence on their performance if they were designing a new user interface (Figure 7.16). They did identify potential problems with one student commenting "*Incomplete set of patterns could hinder design*". Two students also identified constraints, saying "*can be restricting at times*" and "*less room for innovation*"

Communication aid	Count
Very helpful	2
Reasonably helpful	6
Helpful	4
Not very helpful	2
Unhelpful	0

n=14

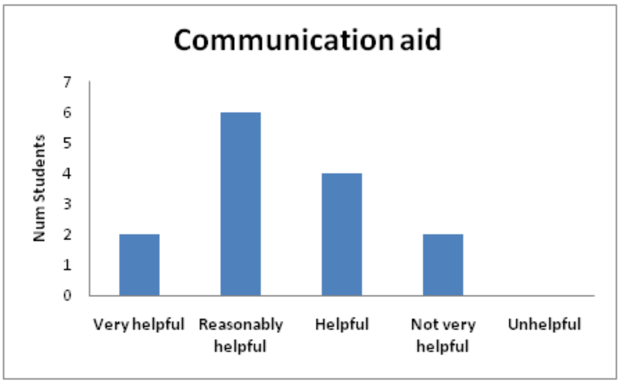


Figure 7.17 – Perceived influence using patterns had on aiding communications

Most students found that using patterns aided communication with their partner (Figure 7.17). One said "*We could compare patterns and argue for or against each one*" which is exactly one kind of communication behaviour a teacher tries to engender. Other comments identified the advantage of having "*a set of standard ways of talking*" or "*without them we would have lacked terminology to discuss the UI features we were looking at*".

7.4 Discussion

Four main methods of data collection have been used to acquire information in this study. Triangulation is achieved by confirming results from two or more forms of data. Multiple sources of evidence add support to any conclusions (Benbasat *et al.* 1987).

The goal for this experiment will have been addressed if the three objectives can be met within the limitations of the experiment. In the following sections the evidence from each set of results will be bought together to determine whether those objectives have been met.

7.4.1 Determine whether UI patterns are an acceptable medium for presenting UI information to students

It is important that the techniques used in teaching aid student learning by encouraging students to think about what they are studying and help them discuss their ideas with their peers (Bonwell & Eison 1991). For patterns to be an acceptable medium for presenting UI information to students not only should the students perceive patterns as being acceptable and useful, but also using patterns should encourage students to participate actively in the assigned activities, to think about and constructively discuss course content. As well the students should be able to use them to complete relevant tasks successfully.

Other researchers (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Chung *et al.* 2004, Segerstahl & Jokela 2006, Saponas *et al.* 2006, Lin & Landay 2008, Koukouletsos *et al.* 2009) indicate that users from a variety of different backgrounds found UI patterns to be helpful. The combined results of the responses to the patterns questionnaires indicated that the student participants in this research are no exception. Students were positive about their experiences using patterns (Figure 7.8). The students agreed that the information presented in patterns was clear, informative and easy to understand although not very easy to remember (Table 7.13) Patterns enhanced and focussed student discussion of the UI modelling task, indicating that not only were the students positive about using patterns but that using them encouraged discussion about user interface knowledge (Figure 7.14). This implies enhancement of learning (Savery & Duffy 2001). That patterns aided their discussion was confirmed by the results of the question in the Exit Questionnaire asking students to rank patterns as an aid to communication. Twelve of the fourteen of the students ranked UI patterns as 'very helpful', 'reasonably helpful' or 'helpful' (Figure 7.17). Students' comments also confirmed this positive ranking.

When asked to consider patterns as an aid to UI design, eighty-six percent of the students considered patterns would be 'very helpful', 'reasonably helpful' or 'helpful' (Figure 7.16). The potential problems students identified in their comments were similar to those identified in the literature (Dearden *et al.* 2002a, b, Chung *et al.* 2004, Segerstahl & Jokela 2006). This indicates that patterns encouraged them to think and reflect on what they were doing. This is supported by students' remarks that patterns helped them focus on the task at hand.

The researcher observed that throughout the tutorial students remained on task and were fully engaged in the process of developing the UI-patterns models indicating active learning is taking place (Prince 2004). This confirmed students' opinions that patterns are an acceptable technique for presenting UI information. A consistent level of discussion occurred throughout the modelling exercises and all discussion overheard covered topics relevant to the tutorial tasks. The images of students pointing to different artefacts and sections within patterns (Table 7.6) along with the movement of artefacts into focus (Table 7.7) throughout the modelling exercise support the researcher's observations that students were fully engaged and on-task during the tutorial.

For patterns to be an acceptable medium for students to use as a learning tool the students should be able to use them to complete UI-oriented activities to an acceptable standard (Kolfschoten *et al.* 2010). In the tutorial, the activity was to create UI-pattern models of reasonable quality for a given user interface and all students succeeded, in both exercises.

7.4.2 Investigate whether illustrated examples influence students' understanding of patterns and pattern languages

Two pattern forms were used to present the information to the students to assess whether the presence or absence of illustrations influenced students use and perception of the information contained in patterns. If pattern form is significant then students should perceive differences in content, layout and organisation of the information. The students were asked to express a preference for either the illustrated or narrative pattern form.

Published results (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Chung *et al.* 2004, Sharp *et al.* 2003) indicate that students were likely to place heavy emphasis on the use of illustrations when using the illustrated pattern form. The frequency of pointing behaviour did not seem to support this finding (Table 7.6). For example, there were no instances of pointing to the illustrated examples at the end of patterns.

Results from the analysis of the items in focus did support the literature (Table 7.7). When using illustrated patterns students are more focussed on matching patterns with the UI, implying that they use a pattern's illustrations to guide decisions rather than pattern content. Matching illustrations to the target UI is an obvious technique for selecting patterns and reviewing decisions. The slightly higher count for list-pattern pairing may be explained by the list marking behaviour (Figure 7.3). Clearly students use illustrations when they are available. The pattern questionnaires indicate that students viewed the illustrated patterns more positively than narrative patterns (Figure 7.11).

The students considered that the information was better organised in the illustrated patterns, although except for the illustrations, the pattern layouts were identical. This anomaly reinforces the importance students place on illustrations. They also perceived the illustrated patterns to be more informative and easier to understand which supports the view that the illustrative examples are representational pictures (Carney & Levin 2002). When responding to the Exit Questionnaire seventy-one percent of the students indicated that they preferred the illustrated patterns. They considered that the illustrations helped them visualise the meaning of the solutions within a pattern's context.

Comments associated with the question in the Exit Questionnaire about the method for creating the UI-pattern model indicate that the students considered the information in the patterns had improved their understanding of user interface development theory and practice. For one student it helped to make sense of the course work studied all semester. Responses to the set of statements asking about the information presented in the patterns were strongly in favour of the patterns, supporting the responses to statements in the Patterns Questionnaire.

Section	Most Useful	Pointing
Name		
Initial illustration	19%	7%
Context section	19%	7%
Problem section	6%	4%
Solution section	13%	15%
Forces section	3%	11%
Reference section	22%	41%
Discussion section		15%
Examples	19%	

Table 7.16 - Comparing pointing behaviour and students' view of pattern sections

Table 7.16 compares observed pointing behaviour (Table 7.5) with students' views of usefulness of different parts of the patterns (Table 7.15). Pointing behaviour shows students interacted mostly with the reference section in both forms of pattern. When asked in the Exit Questionnaire to identify the sections of a pattern they considered the most useful, students identified four sections: initial illustration, context, reference and examples. Although students had a strong preference for using illustrations they recognised the context and reference sections were important because these sections contain the information necessary for linking the patterns into a UI-pattern model. While infrequent pointing interaction with the illustrations was observed, the students' views expressed in the Exit Questionnaire indicate that the illustrated pattern form was perceived as useful. Comments indicated that students thought the illustrations "*facilitated understanding*". Selecting the discussion section as the least useful, along with comments by students that there was "*insufficient time*" and "*too much*" to read, confirmed the findings of other researchers (Wesson & Cowley 2003, Chung *et al.* 2004, Segerstahl & Jokela 2006).

Observations of students' behaviour when working with the two versions of TUI show that there are small but noticeable differences in the way students used the patterns. Students successfully compared patterns (Table 7.7) and interacted with different sections within the patterns (Table 7.6). More pointing behaviour was observed when students worked with the narrative patterns.

When students worked with patterns for the first time, on Exercise One, there is a clear difference in the time taken to complete model building depending on the pattern form used Table 7.8). For the illustrated patterns students averaged twenty-five minutes to build a reasonably complete UI-pattern model, while those using the narrative form averaged thirty-two minutes. Students working with the narrative patterns spent longer becoming familiar with the patterns and most of these students had not begun drawing a model by the time the students using the illustrated patterns had completed their models. Students using the illustrated patterns did not appear to take time to consider pattern content, and probably over emphasised the illustrations. This is confirmed by comments in the Exit Questionnaire by the twenty-nine percent of students who preferred the narrative patterns (Section 7.3.4). All these students previously used the illustrated patterns in Exercise One. They commented that they had not read the information sufficiently carefully, placing excessive reliance on the illustration to understanding a pattern. These students raised issues similar to potential problems identified by other researchers when novice designers use illustrated UI patterns (Dearden *et al.* 2002a, 2002b).

The difference in time taken to complete the two exercises indicates that by Exercise Two students were more confident using patterns and with UI-pattern modelling. Although Exercise Two was more difficult students completed it faster. When working with the illustrated patterns students had completed modelling on average four minutes faster than when working with the narrative patterns. This indicates they had probably developed a reasonable understanding of the pattern structure, content and the UI-pattern modelling method.

The scores the UI-pattern models achieved suggest that students performed better at selecting suitable patterns for modelling a user interface when using the illustrated patterns. However they performed better at identifying potential links when using the narrative patterns (Table 7.10). Comparing the scores for individual students shows a similar picture with more students performing better at selecting appropriate patterns when using the illustrated patterns but performing better at finding potential links when using the narrative patterns (Table 7.11)

These results support the observations that when using the illustrated patterns many more instances of the UI-to-pattern comparisons were recorded (Table 7.7). The images show that most of the time it was the illustrations that the students were focussing on even though images did not capture them physically pointing to the patterns.

Support for the improved identification of links when using the narrative pattern form is less obvious. Firstly, the frequency of observed pointing behaviour was the same for the reference section regardless of pattern form. Pointing behaviour shows students interacted with all the sections other than the lists of examples when working with the narrative patterns (Table 7.6). This indicates that these students paid more attention to content, explaining why they developed a better understanding of links. Comments in the Exit Questionnaire indicate that some students recognised this behaviour.

When the methods used by the students are compared those students who completed Exercise One with the narrative patterns all used the technique where they marked the patterns they selected on the list of patterns.

		CASES	
		Illustrated	Narrative-only
Exercise one	*Hierarchy building*		6, 7
	List marking		4, 5, 6, 7
	Rough copy	2	4, 5
	Reviewing	2, 3	5
Exercise two	*Hierarchy building*		1
	List marking		
	Rough copy	7	
	Reviewing	4	

Table 7.17 – Pattern form in use when students added a technique into the method

Table 7.17 summarises when students were observed to first use a technique in the UI-pattern modelling process. Most of the students who initiated hierarchy building were working with the narrative patterns in Exercise One. Only the students in Case One initiated hierarchy building while working with the narrative patterns in Exercise Two. Students in Case Six used the technique again with the illustrated patterns. List marking was also initiated by students using the narrative patterns in Exercise One. Case Seven probably copied the technique but the researcher's observations that indicated that there was no copying by any other students. No other instances of communication between adjacent pairs were seen, and the work area of each pair of students had at least one row of desks separating them. These observed behaviours of creating an intermediate representation that is different from the final model are a form of Constructionism (Jonassen *et al.* 1996), and may have helped the students to recognise potential links.

There are consistent differences in the way the students worked with the narrative pattern form compared to working with the illustrative form are:

1. **Transformational representation**: The students who used the narrative patterns during Exercise One developed both the hierarchy building and list marking behaviours.

2. **Items in focus**: Picture-to-picture predominated in the narrative pattern informed modelling while UI-to-picture predominated in the illustrated pattern informed modelling.

3. **Time to complete a reasonable UI-pattern model**: Students working with the narrative patterns took considerably longer to complete UI-pattern models. This time difference was more marked in Exercise One than in Exercise Two.

4. **UI-pattern Model:** Students working with the narrative pattern form found significantly more of the potential links and those working with the illustrated form used fewer inappropriate patterns.

7.4.3 Determine whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model

For the method to have been successful, the students need to have followed it as they created their UI-pattern models. The processes students use can also indicate the quality of the learning taking place. The overall scores for the students' UI-pattern models range from thirty-six percent to eight-three percent with an average of fifty-nine percent, (Table 7.10). These results show that students could use the method to create a UI-pattern model but they were not particularly proficient. The marks for the four individual counts used to score the models indicate that students were reasonably successful in building models that were fundamentally correct with regards to the patterns they selected and used (Table 7.11). One explanation for the improvement in pattern selection between Exercise One and Exercise Two is that students have learnt sufficient about patterns to act more confidently when completing Exercise Two even though the exercise was more difficult.

Observations show that the process the students followed contained minor modifications of the given method, but there was no significant diversion from it (Appendix A19). The techniques of marking the pattern list and hierarchy building were developed when students initially used the narrative patterns (Table 7.17).

The other two techniques students initiated, creating a rough copy and reviewing, did not appear to be related to the pattern form students were using. Self-monitoring by reviewing selected patterns occurred more often when students worked with the illustrated pattern form, possibly because they completed their UI-pattern models faster. Reviewing completed work is part of the process of reflection, which is important activity promoting learning. Wood (2001) says:

> *"reflective activity plays an important role in learning and understanding" (ibid, p289)*

These UI-pattern models contained more appropriate patterns than those built when working with the narrative form. But, it is assumed that the illustrative examples dominated students' thoughts because this process did not result in better linked UI-pattern models.

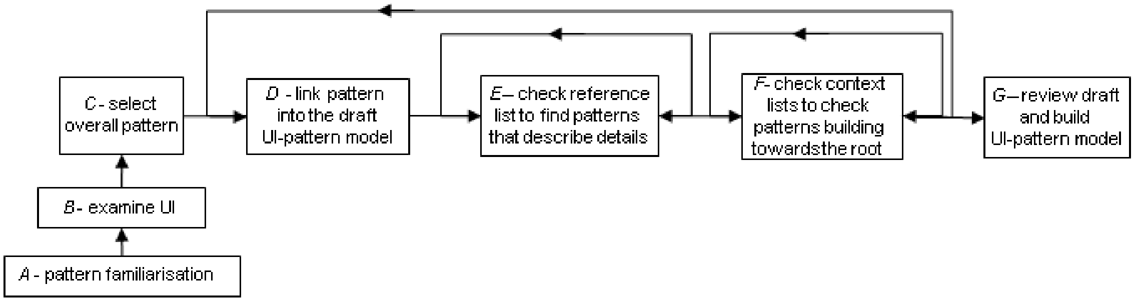
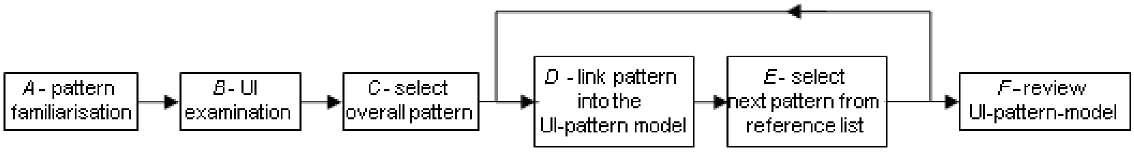


Figure 7.18 – Generalised method combining steps based on students' activities

Figure 7.18 shows a revised UI-pattern modelling method that is a summation of the students' adaptations. It includes two new activities: creating a draft UI-pattern model (Step D) and reviewing, results (Step G). Repetition of tasks in the original proposal is represented with backward pointing arrows. Comparison of the students' UI-pattern models and the links they used, with the TUI pattern map indicates that students made use of both reference and context links. Therefore activities for both have been included (E and F) so that the method covers both top-down and bottom-up approaches to model construction.

The method shown in Figure 7.18 is not recommended for use with students when first introducing them to UI patterns because it is relatively complex and is possibly too prescriptive. Encouraging students to develop their own interpretations of a method can improve their learning experience (Jonassen 1999).



Step A. Become familiar with the subset of UI patterns.
Step B. Examine the user interface.
Step C. Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information the user interface (UI) deals with.
Step D. Link each selected pattern into the UI-pattern model by connecting it with lines showing how it relates to other patterns. When appropriate add a label to the line to match the interaction spaces identified on the UI.
Step E. From the list of patterns that a pattern references, select those that best describe some other feature of the UI. Repeat Steps D and E until no more patterns can be selected.
Step F. Use the context and reference sections of your patterns to check for missing links or patterns.

Figure 7.19 – Recommended UI-pattern modelling method for introducing students to using UI patterns

A simplified version of the UI-pattern modelling method that does not include draft copies and defines only the top-down approach is shown in Figure 7.19. This method should still encourage students to create intermediate or transitional representations.

Responses to three statements in the Patterns Questionnaires related to building the UI-pattern model (Table 7.13) indicate that following the method was a straight-forward process but most students were undecided about the number of steps in the method and whether model building was easy. The responses to the Exit Questionnaire are consistent with these responses. When asked whether the method helped their understanding of the patterns responses were positive. Most students considered the method helpful (Figure 7.15). Comments indicate that with students considering that building the UI-pattern model improved their understanding of patterns and also helped them link UI theory with practical UI design. Suggested changes to the method were mostly concerned with combining some of the steps or rewording them to make the intent clearer. These results support the researcher's observations that the students had no problems following the method when building their UI-pattern models.

7.4.4 Discover whether building a UI-pattern model develops student understanding of UI patterns and pattern language structure

The literature identifies that students initially have difficulty understanding the pattern language structure (Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Wesson & Cowley 2003). By introducing students to UI-pattern modelling it was anticipated that they would readily develop an understanding of pattern language structure. The scores for the UI-pattern models are a surrogate measure (Pfleeger 1994) indicating the extent to which students' understanding of pattern language structure has developed. The students did create acceptable UI-pattern models although there is room for improvement.

The greater difficulty of the second exercise may have had a larger impact on students' understanding of pattern language structure than expected. Table 7.10 shows there was no improvement in the students' performance at finding all the potential links and in the percentage of appropriate links they used in their second models.

Table 7.11 shows that only twenty-nine percent of students performed the same or better at identifying potential links. Only half improved or performed at the same level in selecting appropriate links during Exercise Two. However, more than sixty-four percent of the students' models had over seventy percent of their links correct. Inspecting the reference and context sections of the patterns used in the models indicated that in the second exercise, some students had used all patterns identified in the each pattern's reference section rather than selecting just those that were relevant to the problem. This approach contributed to the poorer link marks in Exercise Two.

The pattern map for TUI that students used had a similar maturity rating to many existing pattern collections (Section 5.4). The anomalies created by unmatched links may have confused some of the less confident students. The structure of TUI had been modified to ensure that if the links between the patterns were followed then finding sufficient patterns to create a complete UI-pattern model was possible. During the modelling process students had to deal with link information contained only within the context section of some patterns. The distribution of the links (Table 7.12) shows that students did identify patterns appropriately by using these links. This can be explained by students actively reviewing their pattern choices. The valid links the students selected when building their models had a similar distribution of link categories as TUI (Table 7.12). The patterns selected were mostly an appropriate subset of the exemplar UI-pattern model. These results indicate that most students had been able to use TUI's structure, but their understanding of it was still developing. Although this evidence is not strong, it supports the premise that some student learning occurred.

The students also introduced technically invalid links into their models, many of which could be classified as missing links. For instance, links from patterns 13 CHOICE FROM A SMALL SET and 14 CHOICE FROM A LARGE SET to pattern 11 TABULAR SET because selections are often a row from a table like structured list. Students found that pattern 15 CONVENIENT ENVIRONMENT ACTIONS needs to be "*used by*" more patterns, for example, 07 MASTER WITH DETAILS and 04 TRANSACTION BY FORM both need to use it. Some of these links are candidates for being added to the participating patterns, the generative process of pattern language improvement (Section 4.2.2).

The students who created the worst models still appeared confused at the completion of the tutorial, but the remainder had developed a satisfactory understanding of both the UI-patterns and the structure of pattern languages. An internally valid version of TUI may overcome such problems.

7.5 Conclusions

The results reported here are from only a small sample of students. These participants are regarded as representative of undergraduate students enrolled in their first in-depth Human Computer Interaction course. Several types of data were collected and triangulation helps increase confidence in the major findings. The analysis of observations and the models students created combined with the responses of the students to both the Patterns Questionnaires and the Exit Questionnaire confirmed that the first two objectives for this experiment were clearly met. Evidence supporting the third and fourth objectives is not as strong but students could successfully use the UI-pattern modelling method from TUIPL to create acceptable UI-pattern models and develop a reasonable understanding of pattern language structure. This experiment has successfully trialled the process of creating UI-pattern models describing user interfaces by student UI developers.

From this exploratory study a number of conclusions can be drawn. First, that UI patterns have proven to be an acceptable medium for presenting information to students. Students found the patterns to be clear, informative and easy to understand. They reported that patterns helped them focus discussion on both the UI modelling tasks and the UI content of the patterns. Observations show they all actively participated in the exercises and discussion. All students built UI-pattern models with varying degrees of success.

Illustrative examples are clearly important for making UI patterns acceptable to students. These illustrations are also important for conveying pattern content and making patterns comprehensible to students. It is concluded that illustrations should be positioned later within the structure of the pattern so that the tendency for students to put excessive reliance on them is reduced.

The UI-pattern modelling method was successfully applied. Students found the method helpful and thought that completing the exercise was straightforward. Students were observed using the method and adapting it as necessary. The method encouraged most students to create transformational representations or draft models as they worked with the patterns. Some students reflected on their work by reviewing their models, while still following the basic process of the method. Both these types of activity can enhance student learning (Bonwell & Eison 1991, Jonassen *et al.* 1996, Wood 2001).

Linking patterns appropriately is difficult, as indicated by the UI-pattern model marks. Three different categories of link made up the pattern map (Table 7.12) with fifty-one percent of the links being unmatched. Students appeared to be confused by this lack of consistency. Analysis of the link structures used in their UI-pattern models indicated that students had a preference for matched links. Consequently, when first introducing students to using a pattern language it should have a valid structure with both the reference and context sections in the patterns written so that all links match.

Students did not gain sufficient understanding of UI-pattern modelling from the tutorial style introduction to start building models without additional help. The introduction demonstrated the use of patterns in general terms with pattern examples and a UI-pattern model based on the architectural domain. Even though they were in their third year of study the students still had difficulty transferring this knowledge into the UI domain.

Finally, it is concluded that developing a UI-pattern model to represent an existing user interface is a successful method for introducing students to using UI-patterns. They were fully engaged in the process, and positive about most aspects related to UI-pattern modelling. Comments show that they found the process improved their knowledge of user interface concepts.

7.5.1 Proposed Changes

The conclusions suggest a number of changes that could be made to improve student learning.

Two changes are proposed to the pattern form used in TUI. First, the size of the initial illustration should be reduced. Second, this illustration should be relocated to the end of the pattern with the other examples.

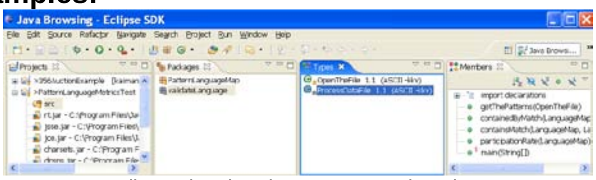
FRONT	BACK
8 - Cascading Collections **Context** The content … **Forces** …. **Solution** *Express….* **Reference** This pattern effectively … **Discussion** …	**Examples:**  Eclipse showing the Java Browsing view: projects>packages>types>members …

Figure 7.20 - Proposed layout for new pattern form

Figure 7.20 shows the proposed layout. These changes will permit students to continue to access the illustrative examples, but should reduce excessive reliance on them and focus attention on pattern content.

It is proposed to rework the primary structure of TUI so it passes the tests for a valid pattern language (Section 4.1.1). Missing links identified by the students will also be considered for inclusion. This will remove any confusion for students when finding patterns, because not all the links in the pattern map matched. The results from scoring the UI-pattern models built by the students were on average reasonable in respect to pattern selection but only just acceptable for the linked structures. An analysis of the individual marks showed that the students were successful in selecting appropriate patterns but a group of students had difficulty finding the appropriate links required to create the model.

Some reference sections in the patterns describe relationships to patterns that define the secondary structure of TUI. These references should be moved to the discussion section to make it clear that they are not part of the primary structure to which the method refers.

An updated version of the UI-pattern modelling method from TUIPL is proposed (Figure 7.19). The final step instructing students to build the UI-pattern model from all the selected patterns was removed. In the revised version, instructions indicate that patterns can be linked into the UI-pattern model as they are selected. The repetition step was combined with the instruction to select the next pattern. A new step was added at the end instructing the students to review the models they have built.

The wording was also modified to make the instructions clearer. For example, some students appear to have been confused when the reference section listed patterns that were not required for completing their model. In the pattern HIERARCHICAL SET the reference section contains the following "*Alternatives to consider are using a sortable TABULAR SET or the CASCADING COLLECTIONS patterns.*" Selecting only one of these two patterns was required. A number of students selected all the patterns listed in the reference section and linked them into their models in the second exercise.

The UI-pattern modelling tutorial used an architectural exemplar. This required students to transfer knowledge gained from the architectural domain into the UI domain. Creating an introduction with a UI-pattern model exemplar would be more appropriate.

Chapter 8: Study Two, Exercise One - Comparison with Study One

This second study implements modifications to TUI, TUIPL and the tutorial introduction that could improve student learning. These changes are based on findings from the first study. The study is divided into two parts. The first part is an iteration of Exercise One used in Study One and is reported in this chapter. Part Two is presented in the following chapter. The aim of this first part of the study was to:

> *To explore whether modifications to teaching techniques and tools influence student learning.*

The objectives for the first part of this study are therefore to:

1. Confirm that UI patterns are an acceptable medium for presenting UI information to students.
2. Confirm that the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model.
3. Discover whether implementing the findings from Study One helped students build better UI-pattern models and helped develop their understanding of pattern language structure.

This study begins with a description of the modifications made to the experimental artefacts based on the results from Study One. The results are then summarised, followed by a comparison with the results for the same exercise in Study One. This comparison is made to determine whether the changes to teaching techniques and tools influenced student behaviour and understanding.

8.1 Research Protocol

At the start of this study the students are introduced to UI-pattern modelling. The research protocol used is similar to that for Study One, Exercise One (Section 7.1).

8.1.1 Experimental Design

Students enrolled in the same HCI course (following year) as described in Study One were invited to participate in this study. Students in both classes had similar backgrounds and both had fulfilled the pre-requisite studies for enrolling in the paper. They were self-selected from the population of students who were likely to enrol in the course at Massey University in any one year. They were the entire population of undergraduate students enrolled in the paper for that year.

This part of the study is a quasi-experiment (Alvarez *et al.* 2006). The introduction to the exercise was altered to use a UI-pattern modelling example. The pattern format and the structure of TUI were updated. Finally, the wording of the UI-pattern modelling method from TUIPL was amended.

The data collection methods (Section 6.3) employed were similar to those for Study One. The data analysis (Section 6.8) aimed to determine whether changes: 'improve', had 'no discernible effect' or 'decrease' student learning.

8.1.2 Artefacts and Activities

The procedures and artefacts used for Exercise One were similar to those of the first exercise from Study One.

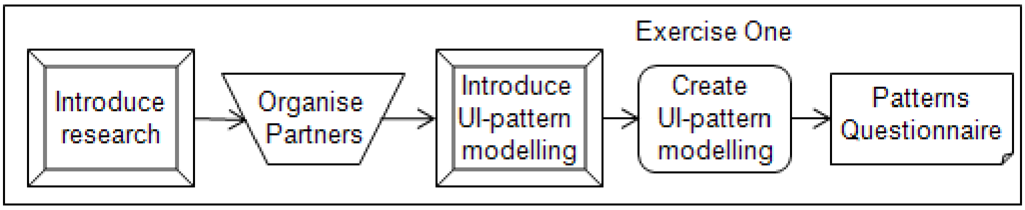


Figure 8.1 - Tutorial procedure for Study Two, Exercise One

Figure 8.1 provides an overview of the tutorial procedures. The planned tasks were followed but the start of the tutorial was interrupted due to several students arriving late. They required individual instruction to catch up with the class.

The artefacts used for Study Two were similar to those of the first exercise in this study (Appendix A6, Section A6.3).

Ethics approval documentation: * Information Sheet * Consent Form
For the UI-pattern modelling exercise: * A version of TUI * A UI example * Instruction sheet outlining the procedures * An answer worksheet for the UI-pattern model * The Patterns Questionnaire

Table 8.1 - Artefacts used by students in Study Two

Table 8.1 lists the artefacts the students used. One difference was that some statements in the Patterns Questionnaire (Appendix A11) were added to take into account the different pattern form being used. The Exit questionnaire (Appendix A12) was administered at the end of the second part of the study but only Section C was directly relevant to this experiment.

Two pilot studies reported in Appendix A3 trialled the procedures and artefacts.

8.1.3 Data Analysis

It was planned to use the same data analysis approach used for Exercise One in Study One (Section 6.8), but due to a camera malfunction detailed narrative analysis was not possible. Actions can be identified but activity counts cannot be used. To compare observations from the two studies the results of analysing the images are discussed by just identifying similarities and differences observed in the way students manipulated the patterns and applied the UI-pattern modelling method from TUIPL.

8.2 Overview of participants

All twenty-four students in the class agreed to take part in the research, but some opted not to complete all questions or tasks. Consequently, respondent numbers are different for different artefacts.

DEGREE PROGRAMME	NUMBER
Bachelor of Arts	1
Bachelor of Business	6
Bachelor of Engineering	6
Bachelor of Information Systems	6
Bachelor of Science	5

n=24

Table 8.2 - Degree Programs students are completing.

Table 8.2 shows the wide range of degree programs in which the students are enrolled. One student was aged under twenty and two older students over thirty, with the remainder in the age range twenty to thirty years. There were seven female students.

Three students reported that they were currently working part-time in the IT industry. A further six reported that they had some relevant IT industry experience. The two students who considered that their proficiency in UI design was very good did not report any industry experience, whereas one student who reported experience in designing user interfaces for games considered their proficiency in UI design was low. The responses to the UI design proficiency question were coded on a scale of one to five. Twenty-two students answered this question. Five students reported they were above average, eight below average and nine considered themselves of average proficiency.

8.3 Key Improvements Identified in Study One

Study One highlighted the need for a series of improvements that could enhance the learning experience for students developing UI-pattern models (Section 7.5.1). Not all changes were used otherwise a comparison between the outcomes of the two Studies would become untenable. The changes implemented were:

1. Reduce the size of the initial illustration and move it to the examples section at the end of the pattern.
2. Rework the patterns to remove the inconsistencies in the link structure making up the pattern map, introduce missing links and rename patterns where student found names unclear.
3. Reword the method's detailed instructions to make them clearer.
4. Modify the introduction to the UI-pattern modelling exercise to use a UI example.

Each of these amendments is discussed in the following section.

8.3.1 Improvement 1 - Modify Illustrations

The patterns to be used in this experiment are similar to the illustrated set of patterns used in Study One, but with the layout modified to reduce the tendency of students to focus excessively on the illustrated examples.

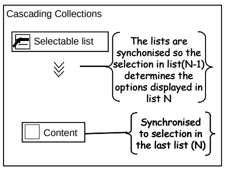
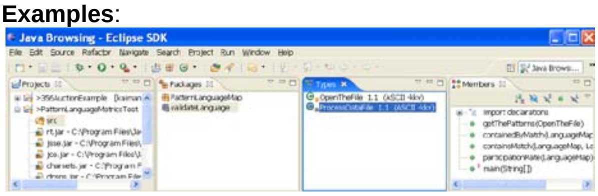
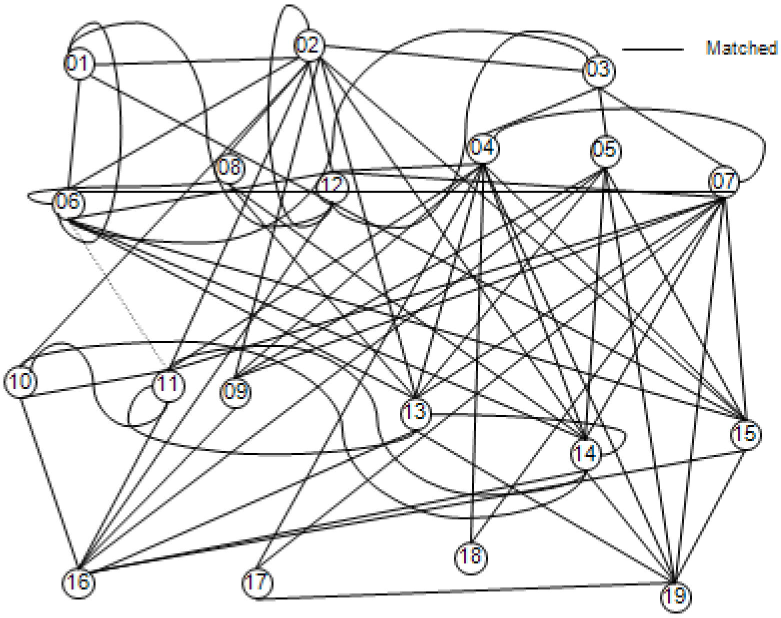
Front	Back
8 - Cascading Collections **Context** The content … **Forces** …. **Solution** *Express ….*  **Reference** This pattern effectively …	**Discussion** **Examples**:  Eclipse showing the Java Browsing view: projects>packages>types>members …

Figure 8.2 – Pattern sections in the diagrammed pattern form

Figure 8.2 shows an example of the diagrammed pattern form which includes a TUIC model diagram identifying the essential elements of the solution. These diagrams are required for Exercise Two. They are relevant to the overall aim of Study Two.

8.3.2 Improvement 2 - Internally Valid Pattern Set



01 – Narrative	11 - Tabular Set
02 - High-density Information	12 - Stack of Working Surfaces
03 – Current Properties	13 - Choice from a Small Set
04 - Information on Form	14 - Choice from a Large Set
05 - Control Panel	15 - Convenient Environment Actions
06 - Collection beside Content	16 - Pointer Shows Affordance
07 - Master with Details	17 - Forgiving Text Entry
08 - Cascading Collections	18 – Ask for Input
09 - Groups with Titles	19 - Good Defaults
10 - Hierarchical Set	

Figure 8.3 - Pattern map for Study Two's version of TUI

The context and reference sections of each pattern have been reworded so that a fully matched pattern map is created as shown in Figure 8.3. As part of this alteration, non-basic references to patterns were moved to another section of the pattern. Three patterns (03, 04 & 18) had names the students did not find helpful. These have been renamed. This remodelled version of TUI is internally valid and its maturity rating has been raised from 14 to 19, i.e. fully mature.

The diagrammed version of TUI passes tests One and Four. To pass Test-two the pattern map in Figure 8.3 has been organised into three levels of detail below the root level. The first three patterns (01, 02 & 03) can be used to provide the overall description of a UI. Each of these patterns can become the root of a hierarchy so the map is effectively made up of a series of overlapping hierarchies. The next group of patterns can be used to describe main interaction spaces within a UI (04, 05, 06, 07, 08, 10). The third group of patterns mostly define smaller areas of a UI that display information with specific characteristics either in layout and/or required actions (09, 11, 12, 13,). The final set of patterns (14, 15, 17, 18, 19) are used for details usually required to model many different parts of the UI so often have multiple links. These form the “leaf nodes” of the structure. To pass Test-three an example is required. The Library example used in the exercises for this tutorial illustrates different levels of model based on the UI-pattern map.

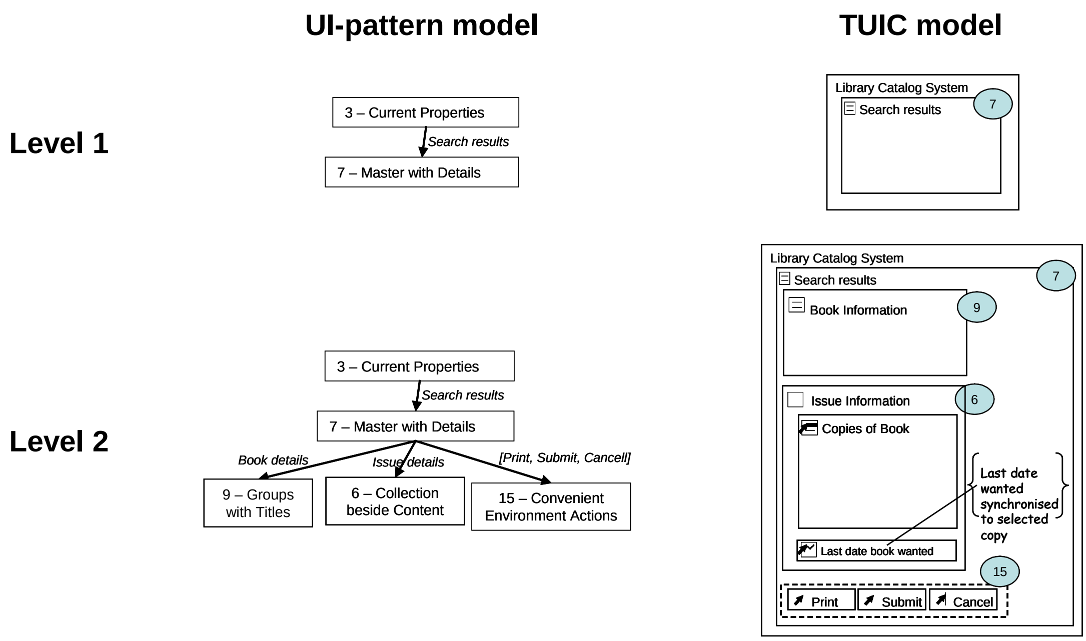


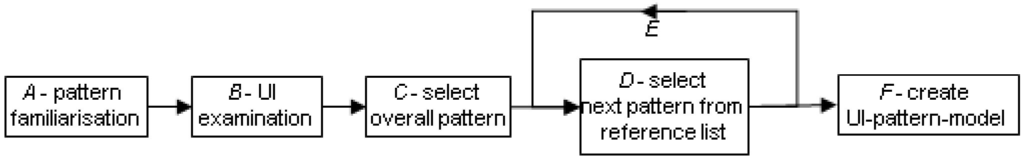
Figure 8.4 – UI-pattern model and matching TUIC model showing two levels of granularity

Figure 8.4 contains two levels of the UI-pattern model and the corresponding TUIC model demonstrates the matching levels of granularity.

8.3.3 Improvement 3 – UI-pattern Modelling Method

The main conclusion from Study One was that the TUIPL UI-pattern modelling method was successful, although an improved version was proposed. The original method encouraged student to initiate the construction of alternative representations which should have contributed to improving their understanding of the material being studied (Savery & Duffy 2001). The researcher was interested to discover whether this behaviour would be repeated, therefore essentially the same method was employed.

Some changes in wording were made where students in Study One had identified problems with wording. Analysis of their’ UI-pattern models also indicated that some may have been confused by the wording. As well, pilot study participants suggested other improvements to the wording.



Step A. Become familiar with the subset of UI patterns.
Step B. Examine the illustration of the user interface for the exercise.
Step C. Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information the user interface (UI) deals with.
Step D. From the list of patterns that this pattern references, select those that best describe some other feature of the user interface you have been asked to model.
Step E. Continue *identifying patterns that* best describe a feature of the user interface *from each selected pattern's reference section* until you can't select any more.
Step F. Using your selected patterns, create a diagram showing your patterns connected with lines to indicate how they are linked into a structure. *Use the context and reference sections of your patterns to check for missing links or patterns*.

Figure 8.5 - Method for creating a UI-pattern model with changes italicized.

The modified wording is shown in Figure 8.5 with changes italicized.

8.3.4 Improvement 4 – Introductory Demonstration

The introduction to the tutorial was modified so that the exemplar demonstrated how to use patterns to create a UI-pattern model. In Study One the introduction used a small architectural exemplar and Alexander's patterns. The replacement UI exemplar was created using the same pattern set the students would subsequently use in their exercises. This change was triggered by observations by the researcher that students who requested help did not appear to understand an explanation based on the exemplar and students' comments that a demonstration based on a UI example was required. When demonstrations are consistent with the expected outcomes they are more effective (Merrill 2002). Although in Study One the process and the model were consistent, the domain was not.

8.4 Results from Exercises One

The students needed to complete Exercise One before Exercise Two because they need to use UI-pattern models to guide TUIC modelling. This experiment therefore provided the opportunity to implement some of the improvements identified from the first study. It provided an opportunity to confirm findings from that study.

The results of analysing the data are summarised in the following sections organised with reference to the objectives. The raw data used for the quantitative analysis are found in Appendix A20.

8.4.1 Observed Behaviours

As with Exercise One in Study One students were organised to work in pairs and are referred to as Cases 1 through to 12. They were encouraged to collaborate while working on the assigned exercise but created individual UI-pattern models. Three objectives similar to those from Study One were used as a basis for these observations, namely:

1. Identify ways the students manipulate the artefacts.
2. Identify sections of the patterns that students focus on.
3. Ascertain whether the students are following the proposed UI-pattern modelling method.

Due to the incomplete sequence of images a detailed case study analysis was not warranted.

Focus of pointing	Cases
Component on UI	3, 4, 5, 11, 12
Exemplar	1
Section in a Pattern	
context	1
reference	3, 5, 6, 7, 9, 12
discussion	3, 8
examples	1, 5, 2

Table 8.3 - Cases where pointing to an artefact was observed

Data in Table 8.3 is from images that capture students pointing to components on the UI and to different sections in the patterns. For example students from Case 1 were seen pointing to the exemplar displayed at the front of the room.

A detailed analysis to count the sections in the patterns students focussed on is not possible due to the incomplete sequence of images. Sections that were captured show that students mostly focussed on the reference section. Only three other sections: examples, discussion and context, were captured being pointed at.

Items in Focus	Cases
Two pattern fronts	3, 8, 11, 12
Two pattern examples (back)	8, 11
Pattern front with UI	1, 2, 3, 6, 7, 8, 9, 10, 11, 12
Pattern example (back) with UI	7, 8, 11
Pattern with list	2, 4
Pattern with graph	5
UI with graph	8

Table 8.4 - Items observed being focussed on by the students

Table 8.4 shows that the majority of items students were observed focussing on were the patterns. They were either comparing two patterns or comparing a pattern with the

UI. In most of the images the patterns were front side up, but for about a quarter they were turned over indicating that when necessary students focussed on the example illustrations.

The students created transitional representation as they worked. One intermediate transformation not observed during Study One was UI marking which three groups in this study employed.

Figure 8.6 clearly shows how the different parts of the UI are identified. These students then appear to review each pattern again when creating their UI-pattern model.

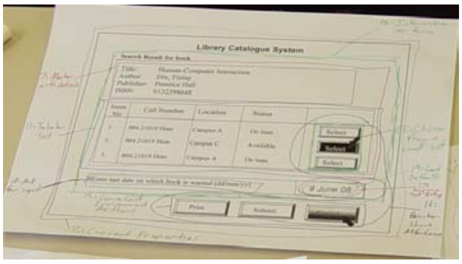


Figure 8.6 – Identifying patterns on the UI

Transformation Model	Cases
Hierarchy building	3, 5, 10
List marking	7
UI marking	6, 7, 11
Rough copy	1, 3, 4, 6, 9, 10, 11, 12

Table 8.5 – Students observed creating intermediate transformation models

Table 8.5 shows that in three cases students were observed moving the patterns into a hierarchical structure to build a UI-pattern model.

In all twelve cases, while creating their UI-pattern models, the students appear to have used a process that is underpinned by the basic structure of the method (Figure 8.5). In ten of the cases students clearly adapted the method by introducing into the process one or more draft model-building activities while completing steps D and E. Three of the teams completed step F in tandem with selecting patterns in steps D and E. In one case students introduced a review activity as they completed step F.

8.4.2 Pattern Model Scores

The objectives underlying building a UI-pattern model to describe the basic components of a UI are to help the students to:

1. Develop an understanding of using patterns to describe a UI component.
2. Develop an understanding of pattern language structure.

The marking scheme is in Appendix A15 and the individual marks for the models are in Appendix A20, Section A20.1.

	max	min	average	standard deviation
Percentage of appropriate patterns	100%	64%	90%	0.11
Percentage potential patterns	90%	30%	60%	0.19
Percentage of appropriate links	100%	33%	83%	0.20
Percentage potential links	89%	22%	54%	0.23
Overall score	92%	47%	72%	0.13

N=24

Table 8.6– Scores for the models created using the diagrammed set of patterns

Table 8.6 shows the scores for all four categories as well as the overall score. Considering the pattern selection and link structures together, the UI-pattern modelling exercise was very successful. The students developed a good understanding of the patterns with forty-six percent of their models containing no inappropriate patterns. Similarly, the students appear to have developed a reasonable understanding of the linked structure of a pattern language with forty-six percent of the models containing no inappropriate links.

8.4.3 Patterns Questionnaire Results

The objectives for the patterns questionnaire were based on those used for the first study plus additional objectives identified from the findings from that study and from specific differences in the pattern form. The objectives copied from Study One are:

1. Find out whether the students considered patterns helpful.
2. Determine students' views about building the UI-pattern model.
3. Indicate whether patterns aided student discussion about UI concepts.
4. Identify whether students developed an understanding of pattern language structure.
5. Determine students' opinion of the accessibility of the information contained in a pattern.

Additional objectives were added for this experiment because all the screen images illustrating the examples are located at the end of the pattern so there is no sensitising image as in the Alexandrian pattern form. The objectives are:

1. Determine students' views of the illustrations used in the patterns.
2. Determine students' views of the pattern names.

The Patterns Questionnaire is found in Appendix A11. The raw data is in Appendix A20, Section A20.3 and the summary data in Section A20.4.

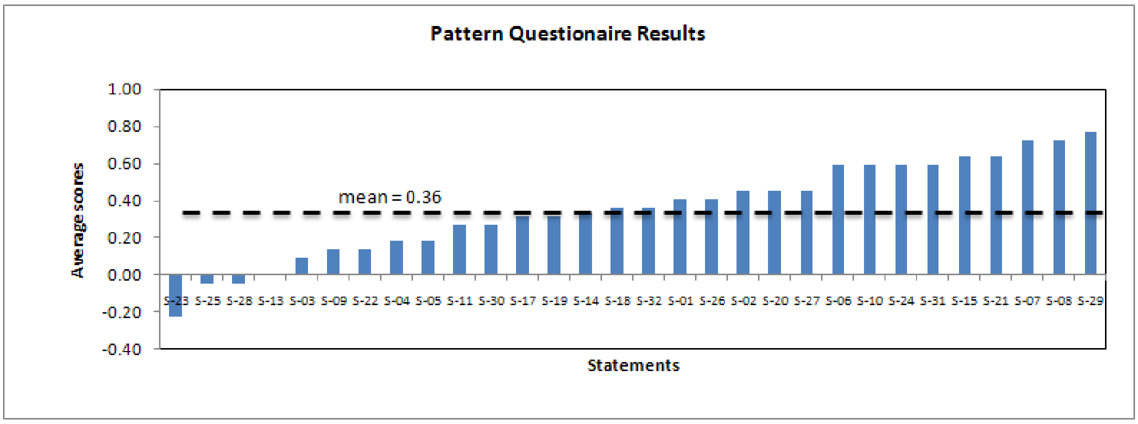


Figure 8.7 – Graph of response totals for each statement

Only twenty-two questionnaires were returned, a response rate of 92%. The overall ranking of the responses to the statements in the questionnaire are positive as shown by the graph in Figure 8.7, with an average of 0.36. This conclusion is confirmed by the results for the statement that explicitly asked whether patterns were helpful. Therefore the first objective is achieved.

Respondent's opinions were elicited for each of six sub-groups of statements corresponding to each of the remaining objectives.

Model building: The results of this set of statements are variable and indicate that the respondents as a group were undecided. Consequently objective two is not met in full.

Topic	S-num	agree	undecided	disagree
Straight forward method	S-14	45%	41%	14%
Number of steps OK	S-23	18%	41%	41%
Method easy to follow	S-28	41%	14%	45%

n=22

Table 8.7 - Responses to statements about aspects of building a UI-pattern model

Table 8.7, shows that forty-five percent of the respondents considered the method was straight forward and forty-five percent did not find the method easy to follow. As a group the respondent students did not appear to agree with the number of steps in the method.

Topic	S-num	agree	undecided	disagree
Aid UI-concepts discussion	S-15	64%	36%	0%
Aid modelling discussion	S-27	64%	18%	18%
Patterns aid focus	S-30	45%	36%	18%

n=22

Table 8.8 - Responses to statements on whether the patterns aided discussion

Discussion: Table 8.8 shows that the respondents clearly considered the patterns helped them discuss both UI-concepts and UI modelling therefore objective three was met in full.

Topic	S-num	agree	undecided	disagree
Straight forward to select pattern	S-03	32%	45%	23%
Finding suitable patterns OK	S-05	36%	46%	18%
Easy to find next pattern	S-10	68%	23%	9%
Referring back to patterns	S-17	55%	23%	23%

n=22

Table 8.9 - Responses to statements on locating patterns within the pattern language structure

Pattern language structure: The results for this set of statements (Table 8.9) are mildly supportive of objective four, that respondents have developed an understanding of the linking structure connecting patterns in TUI. A comparatively large group were still undecided about how straightforward it was to find suitable patterns but most respondents thought that finding the next pattern they needed was reasonably easy. The objective is therefore achieved.

Topic	S-num	agree	undecided	disagree
Clear information	S-01	55%	32%	14%
Patterns informative	S-07	73%	27%	0%
Sufficient information	S-09	45%	23%	32%
Patterns easy to understand	S-18	55%	27%	18%
Logical organisation	S-21	68%	27%	5%
Content easy to remember	S-25	23%	50%	27%
Understand and act on information	S-31	73%	14%	14%

n=22

Table 8.10 - Responses to statements covering information contained within the UI patterns

Information content of patterns: The respondents were very positive that the information was informative and logically organised and that they could act on the information (Table 8.10). Over fifty-percent of the respondents were unsure whether the information in the patterns was easy to remember. This maybe because at most respondents had only forty minutes to interact with the patterns before having to complete the questionnaire. Overall opinions were positive to this group of statements and the fifth objective was attained in full.

Topic	S-num	agree	undecided	disagree
Diagram helpful	S-08	77%	18%	5%
Examples helpful	S-29	82%	14%	5%
Diagram understandable	S-32	55%	27%	18%

n=22

Table 8.11 - Responses to statements about the helpfulness of the illustrations

Illustrations: Table 8.11 shows that respondents considered the TUIC diagrams were helpful and over half considered them understandable. Most of the respondents considered the illustrative examples to be very helpful. Their views are clearly very positive toward both types of illustration so objective six is satisfied in full.

Topic	S-num	agree	undecided	disagree
Pattern names clear	S-19	59%	14%	27%
Pattern names indicate intent	S-26	50%	41%	9%

n=22

Table 8.12 - Responses to statements concerning pattern names

Pattern Names: Table 8.12 shows that the respondents' view the pattern names as acceptable, meeting objective seven. But over a quarter of the respondents did not think the names were clear and a few considered that they did not understand the intent of the pattern from the name.

The results reported above clearly show that the students who responded to this questionnaire found UI patterns to be an acceptable medium for presenting UI information to students. They also found patterns to be clear, informative and easy to understand. They report that patterns encouraged them to discuss UI-pattern modelling as well as UI concepts with their partner. They appear to be confident about using the structure of TUI. They liked the illustrations and even though they had had no specific instructions about the TUIC diagrams they still responded positively to them. Pattern names also appear to be acceptable.

8.4.4 Exit Questionnaire Results

All students completed an Exit Questionnaire but most provided responses to only some questions. The objectives for the section of the Exit Questionnaire covering UI-pattern modelling are to:

1. Investigate whether the students considered the UI-pattern modelling method for introducing patterns was successful.
2. Identify potential changes for improving the UI-pattern modelling method.

The Exit Questionnaire is found in Appendix A12. The numeric data is in Appendix A20, Section A20.6 and the comments in Appendix A22.

UI-pattern Modelling	Count
Very helpful	4
Reasonably helpful	5
Helpful	9
Not very helpful	4
Unhelpful	0

n=22

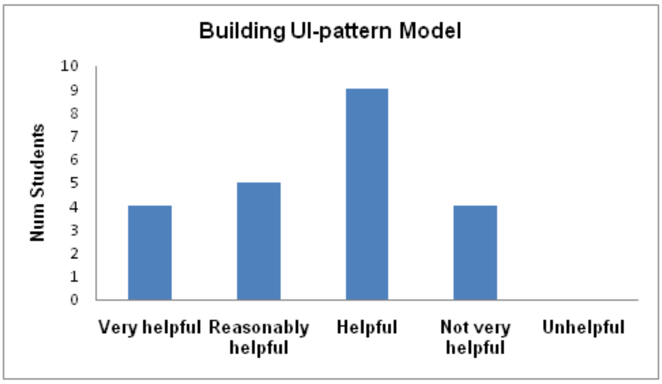


Figure 8.8 – Impact building the UI-pattern model had on understanding patterns

Figure 8.8 show that the majority of students considered UI-pattern modelling a helpful activity. Most of those who ranked the method as 'helpful' or better provided positive comments supporting their ranking. For example, one student commented "*information provided when* [my] *knowledge lacking*" and another commenting that they would "*like to do more*". None of the students proposed any changes for improving the method adding further support for the helpfulness of UI-pattern modelling. The main problem identified by students was lack of time.

8.5 Comparison with Study One's Results

The first exercises in both Study One and Two are essentially identical. They use the same method and example UI for the UI-pattern model building activity. Study One identified a number of issues that require consideration when comparing results between the studies. These are:

1. Students preferred the illustrated patterns but they placed excessive reliance on them.
2. Students working with non-illustrated patterns required longer to complete modelling exercises.
3. Students created transitional models when building UI-pattern models.
4. Students reflected on their activities by reviewing selected patterns while completing their models.
5. Working with patterns encourages relevant discussion.
6. Inconsistencies in the pattern map probably inhibited students developing their understanding of the link structure connecting patterns.
7. Presence of references to additional relationships between patterns in the reference or context sections may have confused some students.

8. A non-UI example in the introduction forced students to transfer knowledge from the architectural domain to the UI domain and may have impeded student learning.

The results of the Patterns Questionnaire, the scoring of the students' UI pattern models and the research observation for both first exercises are compared in the following sections. The purpose is to determine whether the modifications to the introduction, the pattern format and the pattern language structure improve, have no discernible effect or decrease students learning.

8.5.1 Comparing Observed Behaviours

The camera malfunction made a detailed comparison of any modifications to the UI-pattern modelling method difficult. But there is sufficient detail to determine whether the students taking part in Study Two made similar adaptations to those students who took part in Study One.

It is clear that the students participating in Study Two followed the first three steps of the method; becoming familiar with the patterns, studying the UI and selecting a pattern that provided an overall description of a given UI. An analysis of pointing behaviour and the items in focus provide sufficient evidence to conclude that most students were following the information in the reference section to locate the next set of possible patterns. It is not clear how much use was made of the context section, but one instance of pointing to the section was recorded (Table 8.3). As observed in the first study these students created transformational representations as they were building their UI-pattern models. All three forms identified in Study One were observed in this study. An additional form of transformation was also observed. The students identified patterns and wrote the id numbers directly onto the UI. In two cases students were definitely observed reviewing their draft model before completing the UI-pattern model. All these behaviours confirm the validity of the adapted method proposed in the discussion section for Study One (Figure 7.14).

A problem identified in the first study was the tendency for students to place excessive reliance on the initial illustration when there was a large sensitising image on the front page of the pattern. The pointing and comparing behaviour in this second study suggests that students only turned to the illustrations on the back page of each pattern when necessary. The majority of images show students interacting with the narrative content on the front page.

Images of pointing behaviour and items in focus along with the observations of the researcher show that by the end of the exercise students were all on task, fully engaged with completing models and discussing relevant issues.

8.5.2 Comparing UI-pattern Modelling Scores

Scoring the students' UI-pattern models was the primary means for evaluating their learning. Time taken to complete can also indicate how difficult students found an activity. Student's T-test was used to test whether any differences were statistically significant.

	Appropriate patterns		Potential patterns		Appropriate links		Potential links		Overall	
	Study 1	Study 2	Study 1	Study 2	Study 1	Study 2	Study 1	Study 2	Study 1	Study 2
max	100	100	78	90	83	100	71	89	69	92
min	43	64	33	30	42	33	29	22	53	47
average	74	90	55	60	60	83	50	54	60	72
stdev	18	11	14	19	16	20	16	23	4	13
p T<t	0.002		0.41		0.001		0.60		0.002	

Table 8.13 - Comparing scores for UI-patterns models from Study One with those from Study Two

The null hypothesis is that the changes described in Section 8.3 will have no impact on the scores of their UI-pattern models. Table 8.13 shows that the students participating in Study Two performed significantly better overall (p=0.002) indicating rejection of the null hypothesis. Students performance has improved for creating UI-pattern models consisting of appropriate patterns (p=0.002) with on average a sixteen percent improvement. Students also performed better at building models with appropriate links (p=0.001) with on average a twenty-three percent improvement. It is highly likely that the null hypothesis can be rejected for these two criteria as the difference in the marks is significant. There is a minor increase in the average scores for finding all the potential patterns and potential links but these differences are not significant.

	Study One	Study Two
Appropriate patterns	5	6
Inappropriate patterns	2	1
Appropriate links	4	5
Inappropriate links	3	1
	n=14	n=24

Table 8.14 - Comparing the average of counts for appropriate and inappropriate patterns and links

Another way to compare these models is to compare the counts of appropriate and inappropriate patterns and links used in the models. The UI-pattern models created in both studies were on average the same size but as shown in Table 8.14 the models created in the second study contained more appropriate patterns and links and fewer inappropriate patterns and links.

	Study 1	Study 2
% of Models using only appropriate patterns	14%	46%
% of Models using only appropriate links	0%	46%
% of Models using only appropriate patterns & links	0%	33%
	n=14	n=24

Table 8.15 - Comparing the percentage of models created using appropriate patterns & appropriate links

In Study One only fourteen percent of the students created models consisting entirely of appropriate patterns. As can be seen in Table 8.15, in Study Two, forty six percent of the students created models containing no inappropriate patterns. A similar percentage contained no inappropriate links. In Study Two a third of the students created models consisting of only appropriate patterns and links, no students did so in Study One.

	Study 1	Study 2
Range	24-35	25-30
Average	30	28
	n=14	n=24

Table 8.16 - Time taken in minutes to complete UI-pattern modelling exercises

Although the students in this study took time to settle into the first exercise the time taken to complete the UI-pattern modelling exercise was actually shorter than that taken by the students in study one. The range of time taken to finish (Table 8.16) is narrower even though the sample is larger. The students in Study Two on average took two minutes less time to complete the UI-pattern modelling exercise.

8.5.3 Comparing Patterns Questionnaire Results

Because the exercises were essentially the same, it was expected that the respondents opinions concerning pattern use and model building would be similar. The Patterns Questionnaires administered in both Studies included an identical subset of twenty-four statements. The results from this subset were extracted and the rankings then averaged so that the results can be compared using the Chi-square test. All students in Study One and 92% of students in Study Two completed a Patterns Questionnaire.

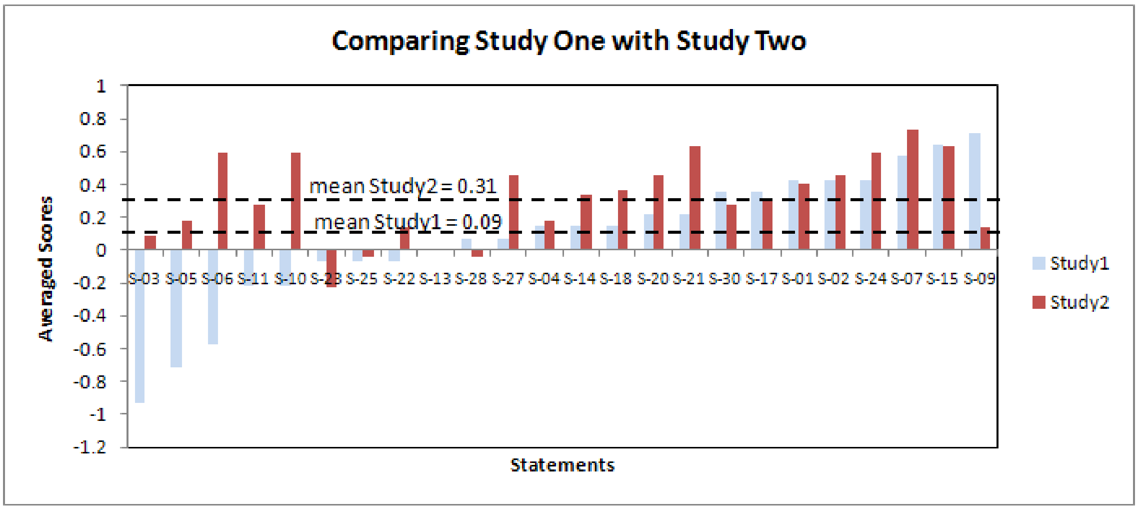


Figure 8.9 - Comparing the rank ordering of responses to the Patterns Questionnaire ordered by rankings for Study One, Exercise One

Figure 8.9 shows the average ranking for Study One was 0.09 while the average for Study Two was much higher at 0.31. The null hypothesis is that the respondents' opinions of patterns would be essentially the same for both Studies. The results indicate that the null hypothesis is highly unlikely with a Chi-square result of 20.681 and a p-value of 0.000 at 2 degrees of freedom. It is clear that the respondents were overall far more positive about patterns in Study Two when compared to Study One.

In the responses for Study Two only three statements register a negative score. Two of these are essentially neutral and similar to Study One. The most negatively ranked statement, S-23, asked the students about the number of steps in the TUIPL UI-pattern modelling method. This statement also had a negative rank in Study One.

The Five lowest ranked statements in Study One were all ranked positively in Study Two. Two of these particularly stand out. Statement S-06 about finding the required information in patterns, and Statement S-10 about how easy it was to find the next pattern, both ranked fourth equal in Study Two. These rankings tend to confirm that respondents were more confident about their knowledge of how patterns are linked together.

The difference seen for Statement S-09 about whether the patterns contained sufficient information was unanticipated. This statement ranks highest in Study One. It was sixth lowest in Study Two. There does not seem to be any obvious explanation for this difference. The three statements that were ranked highest for Study Two all indicated that the respondents considered patterns were informative and logically organised and aided their discussions. Possibly the TUIC model diagrams, which were without explanation, did influence opinions even though the rankings for the statements concerning these diagrams elicited positive responses (Table 8.11).

	Study 1	Study 2
agree	19.05	34.85
undecided	66.67	31.82
disagree	14.29	33.33
	n=14	n=22

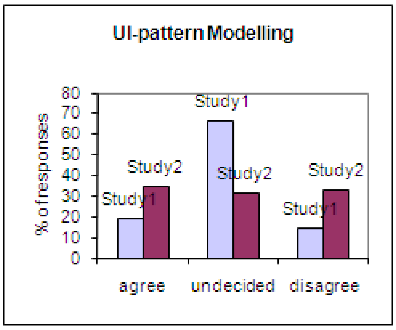


Figure 8.10 - Percentage of counts comparing responses to building the UI-pattern model statements from Study One with Study Two

Accepting the null hypothesis, that there is no difference in the respondents' responses to the model building statements, is unlikely with a Chi-squared result of 12.694 and a p-value of 0.002 at two degrees of freedom. Whereas in Study One most of the respondents were undecided about these statements, responses in Study Two are almost equally divided among the three options as shown in Figure 8.10. These results indicate that at least a third of the respondents considered UI-pattern modelling to be difficult.

	Study 1	Study 2
agree	17.86	47.73
undecided	26.79	34.09
disagree	55.36	18.18
	n=14	n=22

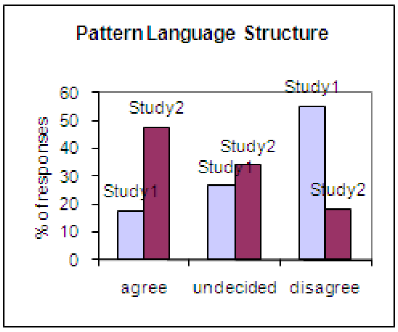


Figure 8.11 - Percentage of counts comparing responses to pattern language structure statements from Study One with Study Two

Responses to the pattern language structure statements indicate that the null hypothesis, that there was no difference between the two Studies, is highly unlikely because the calculated Chi-squared value is 23.530 with a p-value of 0.000 at two degrees of freedom. Figure 8.11 shows the clear difference in respondents' views about using the structure of the pattern language to find patterns. A higher proportion of respondents who participated in Study Two felt they could find the patterns they required.

Students in both studies considered that UI patterns encouraged and informed their discussions. Applying the Chi-squared test to the results for statements covering patterns as discussion aids (Table 8.8) and the information content of patterns (Table

8.10) indicate there is no evidence for rejecting the null hypothesis that the opinions of the respondents were different between the two Studies.

8.5.4 Comparing Exit Questionnaire UI-pattern Modelling Section

The section in the Exit questionnaire asking about the UI-pattern modelling method was identical in both studies. In the first study the students had completed two UI-pattern modelling exercises but in the third study they only completed one UI-pattern model and then created an abstract UI prototype before completing the questionnaire. Comparing the data from the two Studies can provide support for findings from other sources.

The first question asked students whether UI-pattern modelling helped them understand UI patterns. In Study One, one hundred percent of the students responded positively and ninety-three percent provided positive supporting comments. Even though twenty-one percent responded negatively in Study Two, associated comments identified that lack of time was the issue not the method or the patterns. In both Studies the overall response was positive, confirming opinions elicited by the patterns questionnaire. Both groups found UI-pattern modelling helpful in developing their understanding of UI-patterns.

Approximately twenty-one percent of students from both Studies responded to the questions asking for explicit improvements to the UI-pattern modelling method. The responses from the students in Study One addressed specific problems with one student even suggesting alternative wording for Step E. The responses from the students in Study Two were general comments and did not identify specific steps in the method. This difference in responses is probably due to the students in Study One applying the method twice therefore being more familiar with it.

8.6 Discussion

The results from the different data sets related to each of the experimental objectives will be considered in turn. Firstly, to confirm that students found the patterns acceptable and secondly, to confirm the UI-pattern model could be successfully applied and finally to discover whether the changes to teaching techniques and tools improved students' learning. The term 'student' has been used in the following discussion but the reader is reminded that in Study Two only ninety-two percent of the students completed the patterns questionnaire and some Exit Questionnaires were incomplete.

8.6.1 Objective One – Confirm Patterns are acceptable

If patterns are to be an acceptable medium for presenting UI information to students, then the students need to perceive the information they contain as useful and presented in a way that they find acceptable. Criteria considered essential to check that UI patterns are acceptable include whether students became active participants in the UI-pattern modelling exercise, participated actively in discussion and completed their tasks successfully. The quality of the models created can be used as an indicator of students' acceptance of the patterns. The students' opinions about patterns and the modelling method should be positive. They need to believe that patterns helped them with the assigned activities.

At the beginning of the exercise in Study Two some students did not appear to be interested because they were playing games on their cell phones, texting and chatting off topic but they became caught by the exercise as it progressed. The researcher considered all members of the class were actively involved before the end of the first exercise. The results from the observations also show that students actively participated in the UI-pattern modelling activity. They were seen pointing to artefacts (Table 8.3), shifting and marking them (Figure 8.6) and they were also seen comparing patterns (Table 8.4) with each other and with the UI they were modelling. The responses to the patterns questionnaire were overall positive implying indirectly that students were fully engaged in the activities they were evaluating. Active involvement is also indicated by the fact that all students successfully completed a UI-pattern model. Images captured during the exercise show the students comparing items, mostly UI patterns, implying that they were talking about what they were comparing. The researcher observed students actively discussing pattern content and UI-pattern modelling. The students' opinions, as recorded by the patterns questionnaire, were that they found the patterns aided both their discussion (Table 8.8) of UI concepts and UI-pattern modelling (Table 8.9). This positive picture of involvement and acceptance of patterns confirms the situation observed in Study One, although in that study the students were fully involved right from the start.

The overall scores for the UI-pattern models were significantly improved when compared to those from Study One, indirectly confirming that the patterns are acceptable. These differences may due to two pattern types (illustrated and narrative) being used in the first study. Pattern type may be the confounding factor not the changes made to format the patterns for Study Two. Having a better understanding of the UI-pattern modelling process due to the demonstration using a UI example may also account for this apparent time difference. Importantly, the TUIC model diagrams illustrating the solutions do not appear to have had any negative impact on learning about patterns and pattern language structure.

When the totals for all the statements in the patterns questionnaire are considered together (Figure 8.7) the students' opinions overall are favourable towards patterns. An examination of the six subsets of statements in the patterns questionnaire provides a detailed picture of the students' views. These subsets covered: model building, aiding communication, pattern language structure, the information contained within the patterns, the use of illustrations, and the pattern names. For all categories, the overall response was positive towards UI patterns. The positive response for the diagrams used to illustrate a pattern's solution was unexpected because there had been no instruction or discussion about the interpretation of the TUIC model diagrams at this point in the tutorial and no student in the class had any prior knowledge of CAP or navigation symbols used to create them.

Responses to the Exit Questionnaire indicate the student's opinion that building the UI-pattern model method was helpful. This finding is confirmed by responses to the Patterns Questionnaire which indicate that forty-five percent of the students thought the method was straight forward and forty-one percent thought it was easy to follow (Table 8.7). These results also indirectly confirm the acceptability of patterns by students.

8.6.2 Objective Two – Confirm UI-pattern modelling successful

In both Studies the students were presented with essentially the same UI-pattern modelling method from TUIPL to guide the development of their UI-pattern models. For Study Two the wording for steps E and F was improved to make their intent clearer. The criteria for confirming that the method can successfully guide students in creating a UI-pattern model are similar to those used to corroborate the first objective. The criteria are: whether students were fully engaged in the UI-pattern modelling exercise, the models created were of reasonable quality, that students perceived the modelling method as reasonably easy to apply and helpful.

Although the series of images was incomplete it is clear that the methods the students in each Case used were based on the proposed UI-pattern modelling method. In most Cases students modified the method by using transformational modelling activity (Table 8.5) as they selected patterns and in one case reviewed their draft as they completed their final model. One pair of students was observed marking their UI. This type of transformation modelling was not observed in Study One but had been noted by Chung *et al.* (2004) who observed:

> *"A third pair used the patterns in an unanticipated way. Instead of simply culling ideas from the patterns, they annotated their designs with particular pattern references." (ibid, p239)*

Students were clearly fully engaged in applying the UI-pattern modelling method using similar approaches to those observed in Study One.

The results show that the UI-pattern models the students created in this study were of significantly better quality than those created during the same exercise in Study One (Table 8.15). Also they were completed in less time (Table 8.16). These results confirm that the modelling method could successfully be used by the students to create creditable UI-pattern models.

In responses to relevant statements in the patterns questionnaire these students ranked the method more highly that those in Study One (Figure 8.10). Only eighteen percent of the students considered there were too many steps in the method but this was not corroborated by responses in the Exit Questionnaire (Section 8.4.4). The exit questions were administered after students completed Exercise Two, a gap of at least fifty minutes since students had completed their UI-pattern models. This may have resulted in students revising their opinions of the model building process. When asked to identify problems with the method only four students responded and none proposed any modifications. Students identified lack of time as the main problem, with two students wanting '*more explanation*'. Overall these results confirm that the modelling method could successfully be applied by the students.

8.6.3 Objective Three – Changes Improved Learning

In Study One the proposed method appeared to encourage students to construct intermediate or transitional representations as they worked with the UI patterns to create UI-pattern models. Students were seen to reflect on their work by reviewing their UI-pattern models before submitting them, both desirable learning activities (Jonassen 1999). The modifications (Figure 8.5) did not prevent the students in Study Two creating their own representations or reviewing their work. This may have been a factor in the improved perceptions by students in Study Two of how well they could apply the method and the improved UI-pattern models they created. The quality of the models created can be used as an indicator of students' knowledge of UI-pattern language structure.

The UI-pattern models the students created in Study Two were of significantly better quality (Table 8.15). These students completed their models on average two minutes faster even though the start of the tutorial was less than optimal (Table 8.16). Identification of potential patterns and potential links were similar for both Studies (Table 8.13). The ability to exclude inappropriate patterns and inappropriate links was significantly better in Study Two. Over a third of the UI-pattern models comprised only appropriate patterns and appropriate links compared to none in Study One.

Students' opinions about patterns and selecting them were much more positive after completing Study Two when compared to Study One (Figure 8.9). The changes to the tutorial appear to have given students more positive views of UI-pattern modelling and on using links to find patterns. The students who participated in Study Two appear to be more confident in their understanding of pattern language structure (Figure 8.11). The marks for their UI-pattern models confirm this conclusion because the link structures they created were more correct than those of Study One (Table 8.15).

The students in Study One had more experience with UI-pattern modelling and responded more positively about how helpful the method was. They also provided useful suggestions on changes to improve the method. The only comments elicited in Study 2 related to one request for more explanation and two about lack of time which in part was due to the poor start to the tutorial session. The general responses about the method were similar in both studies and indicated students had a positive view of the method.

8.7 Incrementally Improving the Instructional Approach

Underlying the research carried out in this comparison of Exercise One from studies One and Two is the instructional approach which will have influenced student learning. In a design study, as Collins, Joseph and Bielaczyc (2004) indicate:

> *"Each implementation of an educational design is different" (ibid, p34)*

The method of instruction has been incrementally refined by iteration. Problems identified in Study One, suggested the changes that were implemented in Study Two to produce potential improvements. The first part of Study Two, investigates whether the modifications to Exercise One, improved student learning. As already discovered there are known threats to the validity of any conclusions drawn from such a study but the experimental design has minimised these as much as possible.

The results from Study Two clearly indicate improved student learning:

1. A UI-oriented example was used in the introductory demonstration and was available throughout the exercise. This clearly influenced student learning because students were observed referring to it independently during Study Two. This behaviour was not observed during Study One.

2. A validated pattern language structure was used. Results show that it was beneficial to use a validated pattern language when introducing students to using a pattern language. Although there is no direct evidence that using the validated patterns language had more effect than the other changes, students' requests for help with pattern language structure were minimal in Study Two when compared to Study One.

3. The diagrammed form of TUI was used. The changes made to create this version of TUI were: reduced the prominence of the initial illustrated example, restructuring of the link information to other patterns and inclusion of an interpretational illustration in the form of the TUIC model diagram. These all appear to have had a positive effect on student learning.

4. The instructions defining the UI-pattern modelling method were reworded. These modifications may have had little impact on learning outcomes, but removing hurdles to learning, no matter how small, may contribute to student acceptance of an assigned exercise. The responses to the UI-pattern modelling method were neutral and they provided no suggested improvements compared to five students in Study One who responding with potentially useful suggestions.

When scores for the students' models were compared (Table 8:13) there was a statistically significant difference between them. The models created by students in Study Two using the validated version of TUI have the higher scores. Also none of the UI-pattern models created during Study One were composed wholly of appropriate links or patterns, whereas thirty-three percent of those created in Study Two were (Table 8:15). This result is supported by the students' perceptions of how well they could apply the UI pattern modelling method. The perceptions of the students in Study Two were statistically significantly better than the perceptions of the students in Study One (Figure 8:11). Because more than one instructional change was implemented in Study

Two and because of the inherent validity issues in comparing results between students taught in different years no causal association can be proved. But these results are encouraging and indicate further research is merited.

8.8 Conclusion

All three objectives have been met with a reasonable degree of certainty, therefore the aim of this quasi experiment has been met. Clearly the UI-patterns are acceptable, confirming results from Study One and previous studies (Dearden *et al.* 2002a, b, Chung *et al.* 2004, Lin & Landay 2008, Bernhaupt *et al.* 2009, Borchers 2001a, Cowley & Wesson 2005, Koukouletsos *et al.* 2009). The changes implemented in this iteration of Exercise One have definitely had a positive influence on student learning because the resulting models are significantly better than those created in Study One. Due to several changes being implemented simultaneously, conclusions can only be tentative until further research can confirm the outcomes.

After considering all the data from the relevant parts of the patterns and exit questionnaires, and the resultant UI-pattern models, it is clear that the students found UI patterns were an acceptable medium for presenting them with UI information. The results show that the students considered the information in the patterns was presented in an acceptable and understandable format, and that they found patterns to be a helpful aid when completing the UI conceptual modelling exercise. They also agreed that the patterns encouraged and focused discussion with their partner when completing the exercises. The quality of the models created by the students lends indirect support to the acceptability of patterns for presenting UI information to students. Similarly, the students could apply the UI-pattern modelling method and viewed it positively.

The fundamental findings of the first study were confirmed by this study. The students' perception of the task of building the UI-pattern model is positive, and they built well structured models. The process of building the model also helped them develop an understanding of pattern language structure. Incrementally modification of the instructional materials and approach has lead to the improvement in student learning.

Chapter 9: Study Two, Exercise Two - Developing TUIC models

The primary focus of this pre-experiment is trialling the next process in the TUIPL framework. The students are guided through using of a UI-pattern model in the creation of a TUIC model for the specified UI. The aim of this study was to:

> *Explore the use of a UI pattern language in the creation of UI conceptual models by student UI developers.*

The TUIC model combines Canonical Abstract Prototyping (CAP) components (Constantine 2003a) with navigation diagrams (Constantine 1998). They are a type of UI conceptual model. To investigate the aim two objectives have been identified:

1. Determine whether the TUIC modelling method from TUIPL will support students using UI patterns as a guide to the creation of TUIC models.
2. Determine whether UI patterns can be used by student UI designers to gain new UI knowledge.

This exercise follows on from Exercise One (Sections 8.1) of this study. The demographic data describing the participants can be found in Section 8.2. The analysis methods are described in Section 6.8. The results from Exercise Two are presented, followed by a general discussion with particular reference to the research objectives to ascertain whether the research goal has been attained.

9.1 Background

This study investigates primarily aspects of the third of the main research questions (Section 1.3). The students were first introduced to UI-pattern models during Exercise One because they needed to be familiar with these models before they could use them to guide TUIC modelling, the focus of this second exercise.

The motivation for this study originally came from Alexander's statement that "*a pattern must be drawable*" (Alexander 1979, p267) indicating that drawings are fundamental to UI design patterns. The second driver was to introduce students to a UI conceptual modelling method that would help them identify the essential elements of the solution without focussing on specific UI widgets, as found in common tools boxes in integrated development environments (IDE) with which they are familiar. The third driver was to find an aspect of UI knowledge with which students were not familiar with so the premise that students could learn new UI information from UI patterns could be investigated. Using SE design patterns to teach good program design is now standard practice (Shinichi *et al.* 2007, Kolfschoten *et al.* 2010).

A search of the literature found a few general UI pattern languages that used diagrams consistently to illustrate the solutions of the patterns (e.g. Constantine 2003b, de Paula & Barbosa 2003, van Duyne *et al.* 2003, Seffah & Gaffer 2007). TUIC modelling based on Constantine's CAP and navigation symbols was developed for the conceptual model diagrams to illustrate the solutions for TUI (Section 2.3.1.2).

9.1.1 Pre-Experimental Design

This trial is a pre-experiment (Adelman 1991). This type of research design is often used in design studies where it is not possible to use other forms of experimental design. There is no random selection of participants and there is no control for comparison therefore the results are indicative only. This approach is useful for exploratory research and for determining whether further investigation is warranted.

As stated earlier, the entire class agreed to take part in the study but due to time constraints no pre-test was administered. It was known that the students had not been introduced to TUIC modelling and had no prior knowledge of CAP and navigation symbols. They also had no prior experience with the process from TUIPL for creating a TUIC model.

9.1.2 Artefacts and Activities

The procedures and artefacts used for Exercise Two were similar to those of the first exercise in this study.

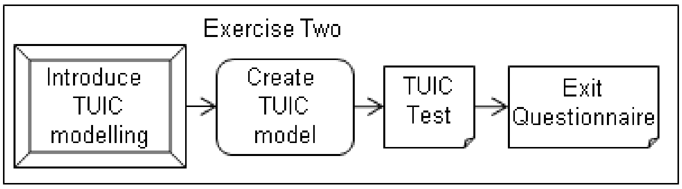


Figure 9.1 - Tutorial procedure for Study Two, Exercise Two

Figure 9.1 provides an overview of the tutorial procedure used for this part of Study Two. The process is similar to that used for Exercise One and starts with a demonstration using the diagrams illustrating solutions in the patterns from the UI-pattern model to create a conceptual UI model using the TUIC modelling components.

For the TUIC modelling exercise: * A diagrammed version of TUI * A pattern list with links listed * A UI example * A UI-pattern model for the UI * Instruction sheet outlining the procedures * An answer worksheet for the TUIC model * The TUIC test
The Exit Questionnaire

Table 9.1 - Artefacts used by students in Study Two

Table 9.1 lists the artefacts the students used for the TUIC modelling exercise. The example UI was the same as used for Exercise One (Appendix A6, Section A6.3). In addition the students were provided with an exemplar UI-pattern model to guide their TUIC modelling (Section A6.4). Providing an exemplar rather than have the students use their own models created during Exercise One eliminates a threat to external validity and makes comparison of the resultant TUIC modelling more reliable. The TUIC test (Appendix A13) was used to determine whether the students had learned about CAP components. The TUIC model method used can found in Figure 5.5.

There is no directed teaching of these components.

The sections of the Exit Questionnaire these students completed included

Section A to collect background information;

Section B to collect data about the patterns;

Section C to collect student reflections on the proposed method used to build the UI-pattern model,

Section D to collect students opinions about the proposed method used to build the TUIC model and

Section E to collect student views on communication with their partner.

9.2 Results from Exercise Two

This research investigated the application of the proposed method for TUIC modelling and also whether UI patterns can convey new information to students. Similar to the previous experiments, four different forms of data collection were used: observation, scoring the resultant TUIC models, a short test covering use of TUIC modelling and an exit questionnaire.

The results of analysing the data are summarised in the following sections.

9.2.1 Observed Behaviours

In each of the twelve Cases (Section 8.4.1) students worked co-operatively on the same assigned exercise to create individual UI-pattern models. A series of digital photographs was taken at regular intervals by the photographer, rotating around the students in the same order during the exercise. The images provide a sequence of snapshots that indicate what the students were doing and the artefacts they manipulating as they completed their TUIC models. The objectives used to guide the analysis of the observation data are similar to those for Exercise One:

1. To ascertain whether the students are following the proposed pattern-guided TUIC modelling method.
2. To identify ways the students manipulate the artefacts.
3. To identify how the students used the UI-pattern model to guide their TUIC models.

The description of each Case is in Appendix A21. The analysis of the image sequences show that the students followed the proposed pattern-guided TUIC modelling method with minor variations. They started by becoming familiar with the UI-pattern model and the patterns describing the UI.

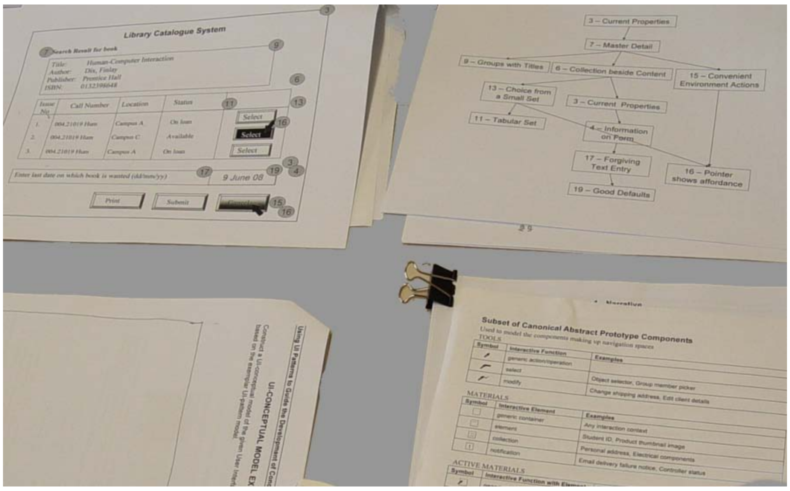


Figure 9.2 - Layout of a workspace showing UI and UI-pattern model side-by-side

In most cases the students started by placing the UI and the UI-pattern model side-by-side on the workspace so both were visible throughout most of the exercise. In some cases they also place the TUIC components list prominently as shown in Figure 9.2.

In two cases (6 & 11) students were observed sorting out the subset of patterns matching the UI-pattern model following Step A. Seven other cases (1, 5, 6, 8, 10, 11 & 12) delayed this step until they were building the lower more complex levels of their models.

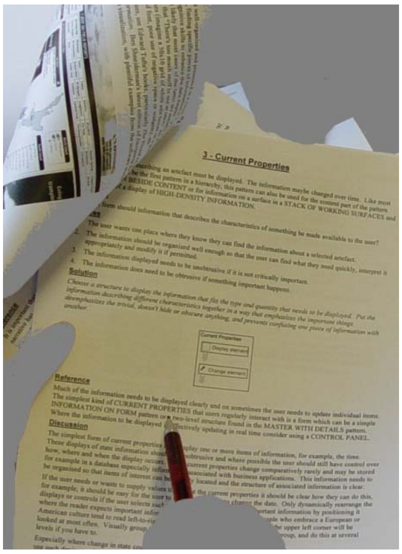


Figure 9.3 - Flipping through pile to locate the required pattern

Most students simply flipped through the nineteen patterns to find a pattern of interest as shown in Figure 9.3. In some cases students appear to have reversed Steps A and B. This behaviour is understandable. To understand the TUIC model diagrams illustrating the solutions for each pattern it is necessary to study the TUIC components first.

The other deviation was to use domain-specific labelling as they worked on their prototypes rather than just using generic CAP symbol labels and updating them to domain-specific labels at the end. Using domain-specific labels appears to have been easier as prototyping progressed. In most cases (3, 4, 7, 8, 9, 10 & 11) the students appear to have only used generic names when they weren't sure what domain name to use. This explanation fits with the fact that a number of completed models still had some generic labels (cases 2, 5, 6 & 12)) even though there was evidence of rub marks showing some names had been updated.

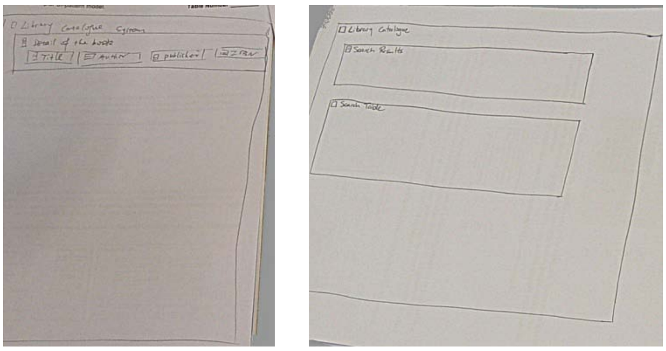


Figure 9.4 - Images showing a depth-first approach compared to a breadth-first approach to TUIC modelling

Two clearly different approaches to TUIC modelling were observed as shown in the Figure 9.4. Some students tended more to a depth-first approach while others used a breadth-first approach. The depth-first group completed all details for each of the three main interaction spaces in turn. The breadth-first group added a container symbol for each of the three main interaction spaces first, before returning to completing lower level details. More interaction with the UI-pattern model is seen as the students completed Steps D and E. They clearly used the UI-pattern model and the annotated UI to guide the selection of the patterns required so they could complete the next stage of their TUIC model.

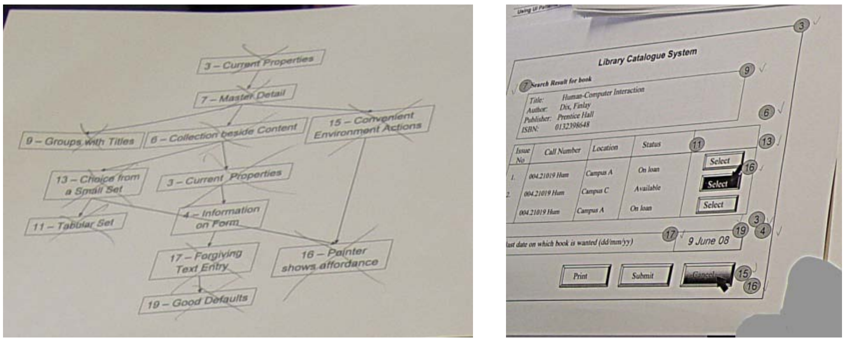


Figure 9.5 - Examples of recording patterns used by marking pattern nodes or pattern ID numbers

As shown in Figure 9.5 the other use made of the UI-pattern model was to mark off pattern nodes to track progress as TUIC modelling progressed (cases 3, 5, 7, 8, & 11). Three Students were also observed keeping track of progress on the UI (cases 3, 6 & 9). Some students (cases 4 & 11) opted to create draft TUIC models on scrap paper before transferring the completed model to the tutorial worksheet (Appendix A7).

Although there was some pointing captured in the images most focus behaviour was in the form of laying a number of artefacts next to each other on the workspace.

Case	Annotated UI	UI-pattern model	Patterns	TUIC model	TUIC components	Other
1	6	1	4	6	1	
2	6	5		5	5	
3	5	5	7	4	1	
4	3	4	5	4	4	2
5	6	3	5	7	1	
6	3	4	4	5	2	
7	4	5		3	4	
8	6	6	3	5		1
9	4	4	1	6		3
10	3	4	6	3		
11	4	4	4	6	1	
12	5		6	5	1	

Table 9.2 - Count of items focused on during TUIC modelling exercise

The collated focus counts organised by Case are shown in Table 9.2. These counts are indicative only. None of the images captured students in Cases 2 and 7 focussing on the patterns even though it is clear they had been moved between successive images. Images capturing students focussing on pattern content was less frequent than expected but this may be because of the cramped conditions that restricted the photographer, resulting in limited views of the students' work area.

Later, two images were taken of each workspace to give a wider view and these often show the relevant patterns positioned to one side rather than on top of the UI and UI-pattern model. These were normally positioned between the two students as shown in Figure 9.2.

Two main activities can be identified. Firstly, using the UI-pattern model to identify the next pattern to select. Secondly, assessing information, presumably trying to understand the TUIC model diagrams and how they relate to components seen in the exemplar UI.

The number of times the students were captured actively working on or apparently discussing their TUIC models also indicates they were fully engrossed in the exercise.

The students clearly followed the proposed pattern-guided TUIC modelling method. As they worked they were mostly observed comparing the different artefacts and were seen to be regularly referring to the UI-pattern model. For example some students marked off the patterns they had modelled as they worked on creating their TUIC models.

9.2.2 TUIC models

The objectives for building a UI conceptual model using the TUIC modelling approach from TUIPL are to help the students:

1. Discover whether they could use the information contained in the patterns defined by the UI-pattern model to create reasonable UI conceptual models.
2. Develop an understanding of TUIC modelling using TUIC components.

The marking scheme is in Appendix A15 and the individual marks for the models are in Appendix A20, Section A20.2.

The students had had no experience with creating TUIC models and only an hour's exposure to the TUIC components and diagrams before creating their models.

	No. 100% correct	min	max	average	standard deviation
Percentage of appropriate TUIC components	3	56%	100%	79%	0.13
Percentage potential TUIC components	0	42%	83%	60%	0.13
Percentage of correct nesting	7	33%	100%	81%	0.17
Percentage of appropriate labels	2	45%	100%	83%	0.14
Overall Score	0	44%	96%	76%	0.14

N=24

Table 9.3 – Scores for the TUIC models created using the exemplar UI-pattern models

The scores the students' TUIC models gained are high, with an average of seventy-six percent (A) as shown in Table 9.3. The averages for three of the four parameters used to score the models are also very high being seventy-nine, eight-one and eighty-three percent respectively. All but one student gained an overall score greater than sixty percent and five students gained an overall score of ninety percent or better. Of these five students, two scored one hundred percent for their labelling, three gained one hundred percent for the percentage of appropriate TUIC components used and the nesting in all five models was completely correct.

(A) **(B)**

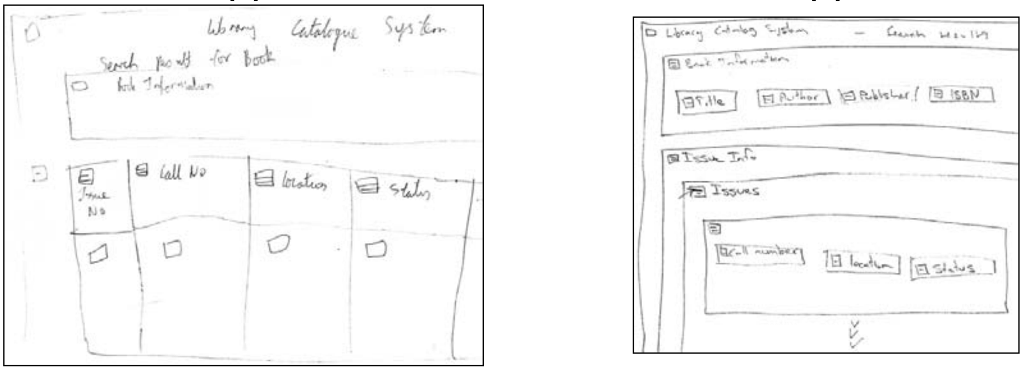


Figure 9.6 - Representations of the collection of information

The students worked cooperatively with their partner. Half of them chose to submit equivalent models but even where they created different models the basic structures were similar. One group of students used a very realistic representation (Figure 9.6 - A) and another used the correct abstract representation (Figure 9.6 - B). In two cases students used a mixed representation.

	Visual representation	Mixed	Abstract representation
Cases	1, 3, 5, 6	10, 12	2, 4, 7, 8, 9, 11

Table 9.4 - Representation of the table structure used in TUIC models

Table 9.4 identifies the Cases using the distinct styles of representation. This interesting result probably shows that even though the students had a correct exemplar diagram to follow the strong visual stimulus of the UI example dominated their thinking.

These results demonstrate that the students used the information from the patterns to create good quality TUIC models. They developed sufficient understanding of TUIC modelling using TUIC components.

9.2.3 TUIC Components Test

The objective for this short test was to:

1. Discover whether the students had developed an understanding of some of the TUIC components from just examining examples in the UI-patterns and working with them to complete a simple TUIC model.

The TUIC test is in Appendix A13 and the marks for the tests are in Appendix A20, Section A20.5.

Only the CAP symbols of TUIC components were tested, not the navigation symbols. The students each completed their own test but most chose to collaborate with their partner. Six students did not complete all parts of the test (Appendix A20).

Range	Number of students
0-2	2
2-4	4
4-6	4
6-8	1
8-10	3
10-12	10

n=24

max	12
min	0
average	8.1
standard deviation	4.0

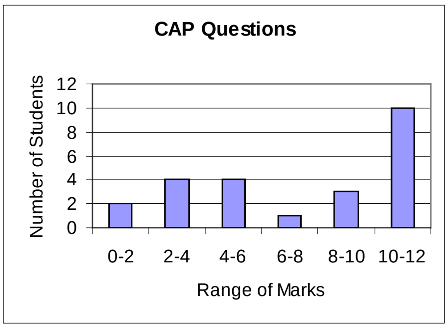


Figure 9.7 - Scores gained in the TUIC test

The results are shown in Figure 9.7. These results indicate that the students developed a basic understanding of using a subset of the TUIC components. It is clear, that weaker students struggled to understand them. Even taking into account that many students worked in pairs the average score is high.

9.2.4 Exit Questionnaire

The Exit Questionnaire contained sections covering both Exercises One and Two. The section covering UI-pattern modelling is discussed in Section 8.4.4. The two sections specific to TUIC modelling and the sections related to both exercises, are discussed below. The objectives for these sections of the questionnaire were to:

1. Identify those parts of the UI patterns that students considered useful.
2. Determine whether patterns helped students communicate relevantly with their partners.
3. Determine whether the method for introducing students to TUIC modelling was successful.

The Exit Questionnaire is found in Appendix A12. The numeric data is in Appendix A20, Section A20.6 and the comments in Appendix A22.

Although responses to the patterns and communication sections by implication covered both exercises they did not explicitly focus students on their experiences in either exercise.

Section	Most Useful	Least Useful
Identifier-Name	3	1
Context	6	
Problem	5	3
Forces	1	6
Solution	2	
Diagram	10	2
Reference	6	
Discussion	3	11
Examples	9	
	n=23	n=21

Table 9.5 - Sections making up a UI pattern

As found in Study One, students have a preference for illustrations. All pattern sections were ranked as most useful by at least one student. Table 9.5 shows that the TUIC model 'Diagram' illustrating the solution was considered the most useful section and the illustrative 'Examples' next most useful. Rankings were supported by the accompanying comments, for the 'Examples' "*helped relate an unfamiliar concept to things commonly encountered*" and for the 'Diagram' "*gave us a quick reference point*". The 'Discussion' and 'Forces' sections were identified as the least useful sections, but still had supporters. Two students commented that they could use the patterns successfully without reading the discussion section. The 'Context', 'Solution', 'Reference' and 'Examples' sections were not considered by any student as least useful.

Diagram helpful	
Yes	19
No	4
n=23	

Table 9.6 – Responses to whether the diagram helped identify essential parts of the solution.

Table 9.6 shows the answers to the question explicitly asking whether the TUIC model diagrams were helpful were very positive, as were the supporting comments. The comments made by two of the students who did not find the diagrams helpful were not particularly negative. For example one said "*The diagrams alone doesn't quite help to get the big picture of what it's* [the pattern is] *about*" indicating that they needed to read the accompanying narrative.

Communication Aid	Count
Very helpful	2
Reasonably helpful	9
Helpful	9
Not very helpful	3
Unhelpful	0

n = 23

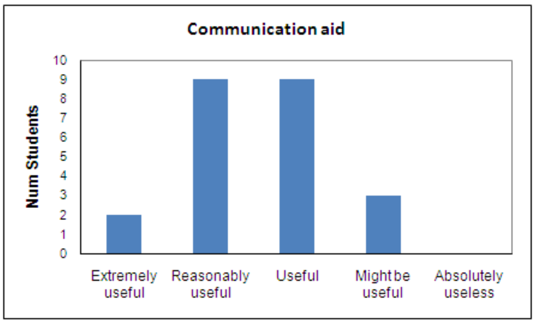


Figure 9.8 - Impact using patterns had on aiding communications

The students definitely considered that patterns helped them communicate appropriately with their partners while completing the modelling exercises, as demonstrated in their responses to the directed question (Figure 9.8). The patterns became a focus for starting relevant discussions and gave some students confidence that they were keeping on topic.

Due to the delayed start the tutorial ran late. In consequence only nineteen students choose to complete the final part of the exit questionnaire which covered the TUIC modelling task.

TUIC modelling	Count
Very helpful	3
Reasonably helpful	6
Helpful	6
Not very helpful	4
Unhelpful	0

n = 19

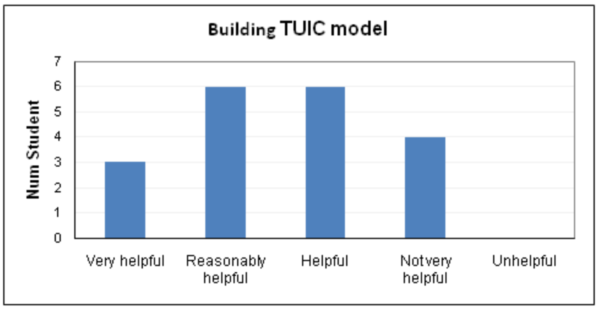


Figure 9.9 - Students' opinions of the influence using patterns had on learning TUIC modelling

Figure 9.9, shows that these students found patterns were very helpful in learning TUIC modelling. Comments indicate that they considered the TUIC modelling method was useful and that patterns increasing their UI knowledge and improved their understanding of UI concepts. None of these students identified any specific problems or offered any suggestions for improvements to the method. One student commented that "*It's clear enough to understand too*".

Patterns were seen to be useful and the majority of comments about the TUIC modelling exercise were positive. One student considered they would be able transfer the knowledge gained, saying "*able to create a UI conceptual model for other user interfaces*". Two other students summed up the whole tutorial by commenting "*Very good tut [tutorial] and very useful*" and "*Enjoyed doing the exercises*".

9.3 Discussion - TUIC Modelling Trial

Results from the data analysis are triangulated and related to the two objectives of this trial. Firstly, the discussion considers whether students could successfully create TUIC models by following the proposed conceptual modelling method, and whether the students learned how to use TUIC components. Secondly, the conclusion consolidates findings to determine whether the aim for the pre-experiment was attained.

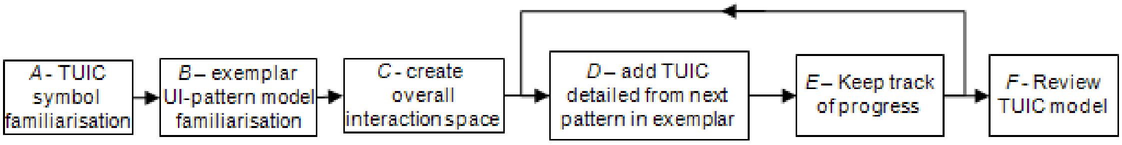
9.3.1 Objective One – TUIC modelling

The second process in TUIPL developing a TUIC model based on the UI-pattern model (Figure 6.3). The data about building these UI conceptual models comes from observing the students while model building, scoring the resulting models and responses to the Exit Questionnaire.

	Reversed Steps A & B	Tracking progress on UI	Tracking progress on UI-pattern Model	Creating Draft	Reviewing TUIC model
Case	2	3, 6, 9	3, 5, 7, 8, 11	4, 11	1, 2, 4, 5, 7, 10, 11, 12

Table 9.7 – Summary of modifications students made to the TUIC modelling method.

The analysis of the sequences of images taken while students completed their TUIC models indicated how the students applied the proposed method. Table 9.7 summarises the changes the students made to the proposed TUIC modelling method. Based on these observations a modified method has been proposed.



Step A. Become familiar with the TUIC components and the UI specification.
Step B. Find the subset of UI patterns used in the UI-pattern model and become familiar with how they describe the example UI.
Step C. Starting with the top pattern, examine the solution and the TUIC model diagram for an overview of the model you are to create. Draw an outline space and label it to reflect the UI domain.
Step D. Work down the UI-pattern model using the TUIC model diagrams illustrating each pattern to guide your modelling. Use lower level patterns to refine the model. If necessary use generic CAP symbols and names otherwise use specific CAP symbols and domain-specific labels.
Step E. Keep track of your progress by marking pattern nodes used, on the UI-pattern model. Repeat from Step D till finished.
Step F. Review your model. Refine TUIC components and ensure labelling reflects the UI domain.

Figure 9.10 - Updated method including tracking progress and reviewing

Students apparently had no need to discard the non-relevant patterns as suggested in Step 1. One possible explanation for this behaviour is that having just completed creating their own version of the UI-pattern model using the same set of patterns, they felt sufficiently familiar with the patterns to not need to sort them at the start of the exercise. Another explanation is that because the pattern set contained only nineteen patterns, leafing through the full set to find a specific pattern was reasonably easy (Figure 9.3). As their TUIC models developed most students extracted a subset of patterns to work with.

Figure 9.10 shows that the modifications based on observations results in a method that remains essentially the same as the original. Although familiarisation was initially directed towards understanding the UI-pattern model, exchanging Steps A and B is very reasonable for those students thinking ahead to creating a TUIC model. Because each pattern has a TUIC model diagram studying the TUIC components should improve interpretation of each pattern's solution while examining the UI-pattern model. An additional directive was added to this step; to become familiar with how the patterns describe the example UI. This was not explicitly stated in the original method. This new wording emphasises reviewing the UI-pattern model and associated pattern content.

The wording for Steps C and D are similar although combined with the old Step E. Additional phrases have been added directing the students to use UI domain-specific labelling because this is what many students did. Minor wording changes refer students to the TUIC model diagram.

Two new steps, E and F have been added. Step E encourages tracking progress by marking patterns on the UI-pattern model. Step F is a review step because in seven cases students were observed reviewing their models.

The method encouraged students to use the UI-pattern model to guide their selection of the next pattern required to create their TUIC model. The students needed to consult the TUIC model diagrams in each pattern, therefore it was expected they would find these diagrams most useful. From analysis of the images of students working on this exercise it was clear that students focus on both the annotated UI and the UI-pattern model. The observations and students self-reporting indicate that patterns aided the student's communications with their partner. Clearly the students were fully engaged with the learning experience and the TUIC models they created were very good. The average score was seventy-six percent (Table 9.3) with a range of forty-eight percent to ninety-four percent.

None of the students identified any problems with the steps in the process they were asked to follow when creating their models. Five students who commented on the method indicated that they found the process clear, enjoyable and useful. Two indicated that they had difficulty with the limited time.

9.3.2 Objective Two – Learning

To be able to determine whether UI-patterns could be used by students to gain new UI knowledge it was important to establish:

1. That the type of knowledge was unknown to the students.
2. That the knowledge was suitable to embed within the current course work, could be learned in a relatively short period of time (less than 100 minutes) and was relevant to UI design.
3. That assessment tools could be used to establish whether learning had taken place.

Two assessments were used. The TUIC models built by the students tested whether the students had learned to apply the TUIC modelling method. The TUIC test assessed whether the students understood and had learned how to use the notation, specifically the CAP symbols.

Responses to the question in the exit questionnaire confirmed that none of the students had had any prior exposure to the TUIC components, CAP and navigation symbols. Constantine says that the CAP notation was created to be

> "*easy for even inexperienced designers to interpret the diagrams and infer the meaning of the notation*" *(Constantine 2003a, p4)*

The students' responses confirm this belief because they found TUIC modelling was relatively easy to understand. In the first exercise students had their attention drawn to the TUIC model diagrams illustrating each pattern's solution, although no instruction was given on how to interpret these diagrams and the key to the symbols used in the diagrams was not available. However fifty-two percent of the students reported that they could understand these diagrams (Figure 8.8) and an even larger proportion (78%) of the students agreed that the diagram helped them identify the essential parts of a pattern's solution (Table 9.6).

The second exercise was designed so that students would use knowledge gained from studying TUIC model diagrams illustrating each pattern to create their own TUIC model of the existing UI. Scores for the students' TUIC models (Table 9.3) indicate whether the students gained a working knowledge about creating TUIC models. The scores averaging seventy-six percent with a range of forty-eight percent to ninety-four percent, confirming that most students gained a good understanding of TUIC modelling.

After the modelling exercise the students completed the TUIC test. Results for this test were also very good (Figure 9.7). Thirteen of the twenty-four students scored ten or more out of twelve.

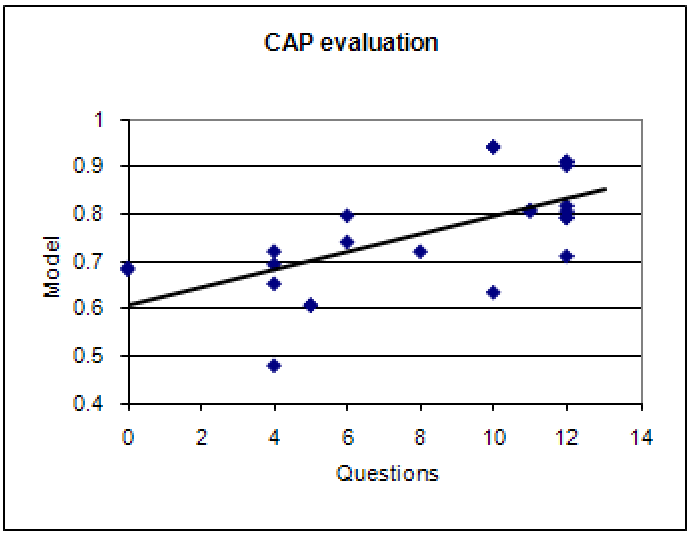


Figure 9.11 - Scatter graph of correlation between TUIC model scores and TUIC test questions

Although the sample size is small (n=24) the results of scoring the models were compared to the scores obtained in the test. Figure 9.11 is a scatter graph of this comparison. The relationship is positive with a correlation coefficient of 0.65 which is significant at (p =0.001). The null hypothesis that there is no correlation between the two sets of marks is rejected. These results are only indicative because students were encouraged to cooperate while building TUIC models and some students also cooperated when completing the test. Finally, further support comes from the students' perception that they had learnt about TUIC modelling from using UI patterns.

9.3.3 The Aim of this Pre-experiment

Given the major limitations of a pre-experiment, the positive results presented above are only indicative. It is clear that these students could successfully follow the TUIC modelling method from TUIPL. The students created very good TUIC models and their opinions were that the method was relatively easy to follow and they also considered UI patterns were helpful. The results from both the test and models show that students had developed an acceptable understanding of the TUIC notation and gained sufficient knowledge to apply the notation to create creditable models. Therefore it is reasonable to state that the aim of this pre-experiment to investigate the use of a UI pattern language to guide the creation of UI conceptual models by students has been satisfied.

9.4 Discussion – Research Question Two

The two studies (Chapters 7, 8 & 9) reported so far comprise the first iterations of the Design study. These two studies explored aspects of how to assist students learning *about* and *through* UI pattern languages. They are directly investigating student learning and the second research question is entirely concerned with teaching students. In the following discussion the results from the three experiments are triangulated to assess whether the second research question has been satisfactorily answered.

Question Two: Are there any specific requirements of a UI pattern language when used for teaching student UI designers?

To answer this research question not only does the impact of pattern language structure on student learning have to be addressed but also aspects of the structure and content of the UI-patterns forming the pattern language. The answer requires evaluating whether students can learn new knowledge by using the pattern language and elucidating their opinions of UI patterns as a way of presenting UI knowledge.

Study	Objective	Observations	Models	Pattern Questions	Exit Questions
1	1. Determine whether UI patterns are an acceptable medium for presenting UI information to students	Yes	Indirectly.	Avg.	Yes
2	1. Confirm that UI patterns are an acceptable medium for presenting UI information to students	Yes	Indirectly	Avg.	Yes
1	2. Investigate whether illustrated examples impact students understanding of patterns and pattern languages		T-test	Chi^2	Yes
1	4 Discover whether building a UI-pattern model develops student understanding of UI patterns and pattern language structure.	Indirectly	Indirectly Correlation	Indirectly	Indirectly
2	3 Discover whether implementing the findings from Study One helped students build better UI-pattern models	Indirectly	T-test	Indirectly	Indirectly
				TUIC Test	
2	4. Determine whether UI patterns can be used by student UI designers to gain new UI knowledge	Indirectly	Avg.	Avg.	Yes

Table 9.8: Overview of findings from Studies One and Two that support the second research question

Table 9.8 lists the objectives from Studies One and Two that relate directly to the second research question. The table provides a summary of the findings from the four types of data collected. Where the results indirectly support the objective the findings have been prefaced with the word ‘indirectly’. Where a quantitative measurement was made the type of test used, i.e. t-test, chi-square or averages have been indicated.

All the evidence that the students’ developed an understanding of the structure of a pattern language is indirect. Also the activity of building the UI-pattern model and the structure of TUI will both influence whether students’ develop a satisfactory understanding of the pattern language’s structure. These influences cannot be separated. Furthermore, because more than one modification was implemented between the two studies no one modification can be directly credited with improvements in student performance.

9.4.1 UI patterns are acceptable

To meet the first objective for Studies One and Two (Table 9.8, rows 1 & 2) students need to have agreed that UI patterns are an acceptable form for presenting UI information. They should also have been able to use the patterns in a constructive manner as demonstrated by creating UI-pattern models and TUIC models.

The evidence for UI patterns being an acceptable medium for presenting UI information to students comes from responses to the Pattern Questionnaires statements, which were very encouraging (Figures 7.7 & 8.7). Students’ opinions collected in the Exit Questionnaires also indicate that patterns were an acceptable method for presenting UI information, especially in aiding communication when working collaboratively (Figures 7.17 & 9.8). This evidence confirmed the findings of a number of previous studies (Borchers 2001a, 2002, Cowley & Wesson 2005, Koukouletsos *et al.* 2009). Koukouletsos (2008) summarises the pedagogical value of UI patterns identifying them as organising and structuring knowledge in an accessible format thereby making that knowledge easy for students to understand, learn, remember and apply. In this research, students were observed to become fully engaged in both the UI-pattern modelling exercises and the TUIC modelling exercise. Students responded positively to using the UI patterns in the modelling exercises they completed (Figures 7.16, 8.10 & 9.9). Observational data showed that students used the patterns actively to solve the modelling exercises, with images showing students consulting, pointing to and comparing patterns (e.g. Tables 7.6, 7.7, 8.3 & 8.4). Transformational model building activities also indicated that students were actively engaged in using the patterns (Tables 7.17 & 8.5).

In studies One and Two, the average score for the Exercise One UI-pattern models were 60% (B) and 72% (B+) respectively, indicating that students could use the UI patterns reasonably successfully. Stronger evidence for patterns being acceptable comes from Study Two where thirty-three percent of the UI-pattern models created consisted entirely of appropriate patterns and links. The average scores for the TUIC models created during the second exercise in Study Two were also high at 76% (A-). Other researchers (e.g. Dearden *et al.* 2002a, b, Finlay *et al.* 2002, Koukouletsos 2008) have also found students can create creditable UI designs when using UI patterns. Clearly students can use UI-patterns constructively, providing evidence that UI patterns are an acceptable medium for presenting UI information to students.

9.4.2 Impact of Illustrations

The literature review indicated that both novice designers and students could place excessive reliance on either the pictures located at the beginning of many UI patterns (Dearden *et al.* 2002a, Chung *et al.* 2004, Sharp *et al.* 2003, Finlay *et al.* 2002) or on the illustrated examples within the body of the pattern (Koukouletsos 2009). This phenomenon is supported by the results of this research. Students who worked with the narrative version of TUI were significantly better at finding the appropriate relationships linking the patterns. Finding links relied on careful reading of the context and reference sections of each pattern. Some students also self-reported they had given undue attention to the illustration.

There is a widely accepted maxim that “pictures are worth a thousand words” but educational research indicates that the type of picture can greatly influence students’ interpretation and understanding of accompanying text (Carney & Levin 2002). In the condensed form of information packaging that comprises a UI pattern, it is important that the illustrations have a positive impact on learning.

The image placed near the beginning of a UI pattern should be sensitising and memorable (Graham 2003). This description seems to fit the transformational type of illustration. Alexander (1979) describes this initial picture as an archetypical example which describes a ‘representational’ illustration (Carney & Levin 2002). According to the classification system described by Carney and Levin (2002), if the selected illustration is representational some improvement in learning should be indicated when comparing the use of narrative-only patterns with illustrated patterns. If the illustration is transformational then it is reasonable to expect that a marked improvement in learning should be observed. The comparison of marks for the UI-pattern models students built in Study One, Exercise One comparing narrative and illustrated pattern forms, were not significantly different. This indicates that the large image of an example UI placed near the beginning of a pattern is not transformational. This image can be classified as representational because the improvement in marks for selecting appropriate patterns was weakly significant when students worked with the illustrated patterns.

The diagrammed version of TUI included TUIC model diagrams illustrating the essential elements of each solution. These diagrams were the only illustration appearing on the front page of the physical layout of the pattern. These diagrams can be categorised as ‘interpretational’ illustrations. Interpretational pictures should improve student learning more than using just representational pictures (Carney & Levin 2002). This new pattern form retained the representational pictures but these were placed at the end of the pattern and reduced in size. Reducing the size of the archetypal illustrative example and moving it, was not observed to impede students’ use or acceptance of the patterns. The UI-pattern models created using the diagrammed patterns in Exercise One of Study Two scored significantly better marks than those created during the same exercise in Study One. This result needs to be treated with caution because the two student groups came from different enrolment years and some unknown factor in the background of each student group may have influenced the results. Similar validity issues are a known problem with Design Studies (Bell 2004). In this research both groups of students satisfied the same set of prerequisites to enrol in the HCI course, the faculty were the same, and the lecture programme and assessments were comparable.

9.4.3 UI-pattern Language Structure

Patterns selected when creating TUI were relatively simple. Any pattern that had the potential to distract students from focussing on pattern language structure was not included. Patterns relating to the behavioural or social aspects of UI design were eliminated because most of these patterns were considered too abstract. Support for this decision is found in research by Borchers (2002) who reported that even graduate students had difficulty with both granularity and abstraction. Koukouletsos (2008) reported that undergraduate students had difficulty with compound patterns compared with simple patterns that related directly to a UI component. Compound patterns seemed to "*require more attention from students*" (ibid, p229) than they were able to give when first using UI patterns. Only spatial and temporal descriptive patterns were retained. These restrictions ensured that the UI-pattern models created using TUI linked into a single hierarchical network.

Observations show that while students were completing UI-pattern models the number and type of questions related to pattern language structure were minimal during Study Two. This behaviour contrasted strongly with that observed in Study One where many questions were asked about how to use the structure of the pattern language, and also about the inconsistencies between context and reference sections of linked patterns. These observations indicate that building the UI-pattern model probably focussed students' attention on the structure of the pattern language. The building of intermediate representations (Tables 7.17 and 8.5) also demonstrated that students were using the pattern language structure when creating their UI-pattern models especially list marking and hierarchy building.

The UI-pattern modelling method used to introduce the students to TUI directed them to use the structure of the pattern language to build UI-pattern models. The strongest evidence that they were actively using pattern language structure comes from the correlation of the link types the students used and those present in the version of TUI they were using for Study One (Table 7.12). This version of TUI was moderately mature (14) which is commensurate with existing real world pattern languages (Figure 4.4). The other evidence is the quality of the UI-pattern models created in both studies. The students in Study Two worked with an internally-valid version of TUI that was fully mature (19). Forty-six percent of these students' models contained no invalid links but in Study One all models contained at least one invalid link (Table 8.15). In this version of TUI only basic relationships were mentioned in the context and reference sections of each pattern, with all non-basic relationships described in the discussion section.

9.4.4 Gaining New UI Knowledge

Seffah (2003) clearly considered UI patterns could support both learning and teaching in an educational setting. He saw UI patterns as a useful tool that captured good UI design practice and could be used to transfer such knowledge to student learners. A number of studies support this view and indicate that students produced better UI prototypes when working with patterns (Laakso *et al.* 2000, Griffiths & Pemberton 2004). Only

Koukouletsos *et al.* (2009) confirmed that the students in his study had no prior usability knowledge. The resultant prototypes were scored on usability features. He found that learning from patterns had occurred, saying:

> *"The results of the experiment indicate that patterns ... can support novice designers in learning and applying usability principles" (Koukouletsos 2008, p272)*

The research reported in this thesis also supports the premise that students can learn new UI knowledge from using patterns. The students in Study Two had no prior knowledge of TUIC modelling but they created good TUIC models with an average grade of A- (Table 9.3). Further analysis showed that if the comments were not included in the scoring scheme then eighty-four percent of the models had less than two missing CAP symbols. The scores for the TUIC test were similarly high with thirty-three percent of the students scoring one hundred percent (Figure 9.7). There was a significant positive correlation (coefficient = 0.65 at p=0.001) between the two assessment methods. The students also reported that they learnt new knowledge from using the patterns and that using TUI helped them learn about TUIC modelling (Figure 9.9). These results support the premise that students can learn new information from working with a UI pattern language.

9.4.5 Specific Requirements of a UI Pattern Language for Teaching

The discussion above provides sufficient evidence to have confidence that when introducing students to UI patterns and having them use UI patterns to guide UI design activities, to help improve their learning outcomes the pattern language used should:

1. Be internally valid.
2. Use a pattern form that is written so only basic relationships are identified in the context and reference sections of the pattern.
3. Use a pattern form that is written so non-basic relationships are identified separately from the context and reference sections, for example in the discussion section.
4. Use a pattern form that includes illustrations of archetypical examples. These should be placed in a less prominent position than the beginning of the pattern form and reduced in size so as not to draw undue attention to them.
5. Use a pattern form that includes diagrams identifying the essential elements of the solution, for example an interpretational illustration such as TUIC model diagrams.

9.5 Conclusion

After presenting the results from the pre-experiment investigating students using TUI to guide the creation TUIC models, a discussion followed to demonstrate that the objectives for this experiment have been met with a reasonable degree of certainty, within the limitations of the pre-experiment process. A number of different data collection techniques, both qualitative and quantitative, were used and the results from the analyses have been triangulated. Due to the very positive results it is reasonable to state that the aim for this pre-experiment has been satisfied.

The data for assessing whether the research aim has been met comes from the observations, the students' TUIC models, the TUIC test and the relevant section of the exit questionnaire. Crucially, it was established that the students had no prior knowledge of TUIC modelling. TUIC modelling is reliant on the availability of a UI-pattern model matching the target UI. The analysis of the images shows that students repeatedly referred to both the UI-pattern model and the annotated UI example, as they selected patterns to complete their TUIC models. The students' perceptions of applying the TUIC modelling method were positive and they considered that the patterns had been useful in helping them learn about TUIC modelling. Students were shown to be able to use the TUIC modelling method from TUIPL to create models of acceptable quality. They learned to use the TUIC components from working with the TUI patterns, demonstrating that a pattern language can be used successfully to teach students new UI knowledge.

This pre-experiment concluded the set of experiments that trialled different aspects of using TUIPL and TUI with students in a class-room setting. The chapter concludes with a discussion answering the second research question that guided this research and a set of requirements for a teaching UI pattern language that should improve student learning outcomes was defined.

Chapter 10: Study Three - Pattern Guided UI Design

This study investigates whether the first two processes in the TUIPL framework can guide the development of a design for a new user interface from user requirements (Figure 5.2). This complements the processes tested in the previous two studies. The investigation consists of three case studies which explore the viability of UI-pattern modelling and the TUIC model with experienced software developers to ascertain their opinions. The unit of study comprises the first two processes in the TUIPL framework along with associated tools and techniques. The study uses a scenario in an exercise situation.

The aim of this study was to:

Explore whether the proposed UI pattern guided design method would provide students with an authentic UI design experience.

To examine this aim the following two objectives were identified:

1. Determine whether UI developers could learn and apply the first two methods in the TUIPL framework to develop conceptual UI models for a new user interface.
2. Determine UI developers' *opinions of the viability of* the first two methods in the TUIPL framework in a *professional* setting.

To meet the objectives each participant needs to be able to comprehend the UI patterns and the structure of the pattern language. They should be able to create a UI-pattern model from the requirements documentation for a new user interface and then use the UI-pattern model to guide the development of a TUIC model for that UI.

10.1 Protocol Design

This study is exploratory and using a multiple descriptive case study approach. It consists of three case studies that examine the pattern-guided UI design method. The researcher acted as a participant-observer. The researcher introduced the participants to the two design methods: UI-pattern modelling and TUIC modelling. During the design exercises the researcher supported the participant's modelling efforts.

After the initial introduction to the research the participants who agreed to take part complete two further sessions. Data was collected using similar methods to those employed in the earlier studies (Section 6.3). Descriptions of the observations, the artefacts created, the data from the questionnaires and ideas from the unstructured discussion that concluded each session.

Results are described as a within-case analysis where each case study is analysed separately. They can be found in Appendix A24. After the within-case analysis a cross-case analysis is carried out (Yin 1994, Flyvbjerg 2006). The cross-case analysis is reported below. It identifies similar patterns of response, comparing and contrasting results of the within-case analyses.

10.1.1 Artefacts and Activities

The procedures and artefacts used for this Study are similar to those used for Study Two.

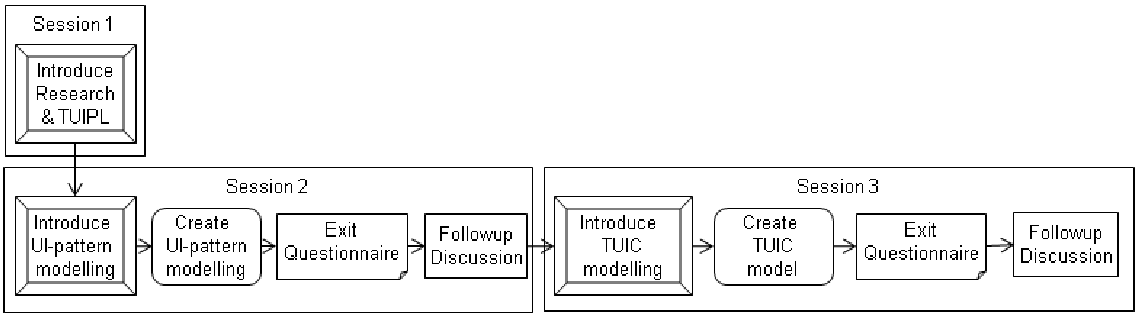


Figure 10.1 - Case study procedure for Study Three (Copy Figure 6.11)

Figure 10.1 provides an overview of the procedures for each of the sessions for the three case studies. The list of the tasks for each of the three sessions is in Appendix A30. The introductions to each of the modelling methods involved the participant reading some material and the researcher demonstrating the actual modelling method using the workbook as a guide. The participant could ask questions to clarify points and could also stop the demonstration while referred to the workbook.

<table>
<tr><td colspan="2">For the introduction:
* Information Sheet
* Ethics Consent Form
* Background Section of questions
* A set of patterns</td></tr>
<tr><td>For the UI-pattern modelling exercise:
* The UI requirements
* Introductory UI-pattern model workbook
* A set of patterns
* A pattern list with links listed
* An answer worksheet for the UI-pattern model
* The Exit Questionnaire</td><td>For the TUIC modelling exercise:
* The UI requirements
* Introductory TUIC modelling workbook
* A set of patterns
* A pattern list with links listed
* A UI-pattern model for the UI requirements
* An answer worksheet for the TUIC model
* The Exit Questionnaire
* The patterns Questionnaire</td></tr>
</table>

Table 10.1 - Artefacts used by participants in Study Four

The artefacts for each of the three sessions are listed in Table 10.1. The relevant sections of the Exit questionnaire for this study were administered separately at the end of each of the three sessions.

The first two procedures in the TUIPL framework are trialled. The detailed steps defining the development of the UI-pattern model and the development of the TUIC model from an existing UI have been modified to describe the development of a new UI. The changes include replacing the existing UI with high-level UI requirements documentation for the new UI. The observations of students working with the simplified versions of these two methods in studies One and Two (Figures 5.4 & 5.5) influenced the development of the full methods.

The methods presented to the students described a simplified top-down approach. The full methods are more flexible so that top-down, middle-out or bottom-up approaches to modelling are all possible (Yourdon 1989). The user is not directed to start by selecting from just the subset of potential root patterns.

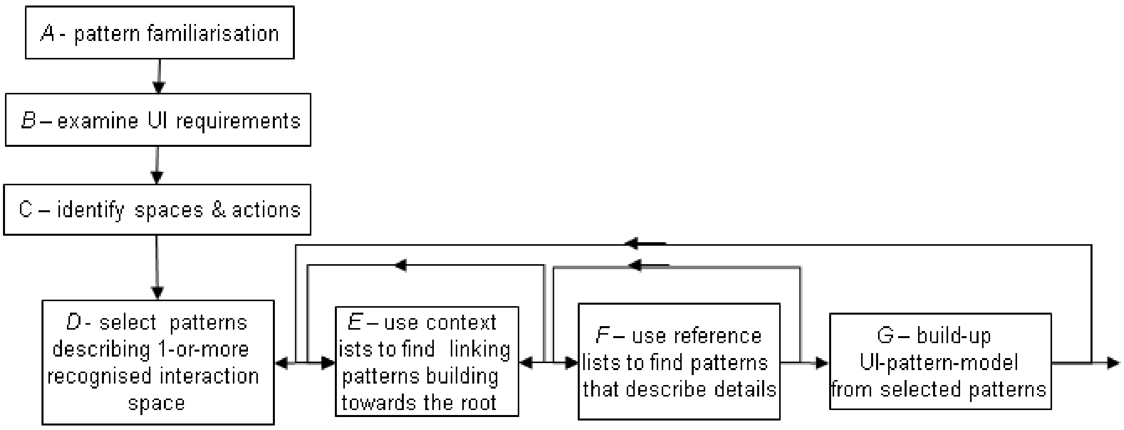


Figure 10.2 - Schematic of the UI-pattern model building method

Figure 10.2 shows the full UI-pattern modelling method. The detailed steps for this and the TUIC modelling method can be found in Appendix A9.

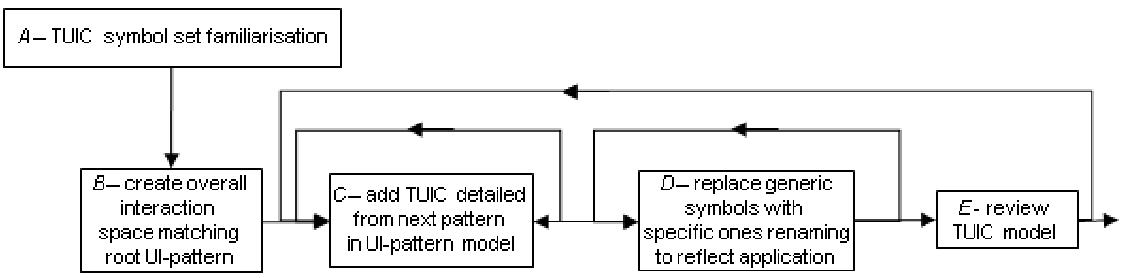


Figure 10.3 - Schematic of the pattern informed TUIC modelling method

Similarly, Figure 10.3 shows that the TUIC modelling method was modified to be more flexible than that used in Studies One and Two. The existence of the related UI-pattern model is assumed.

At the end of the **introductory** session:

- Section A, to collect participant back ground information.

At the end of Session Two - **UI-pattern modelling**:

- Section B, asks about the patterns,
- Section C, asks the participants to reflect on the proposed method used to build the UI-pattern model,
- Section F asks for opinions about UI-pattern modelling aiding non-professionals participate within the design team.

At the end of Session Three - **TUIC modelling**:

- Section B asks about the patterns ,
- Section D asks the participants about the proposed method used to build the TUIC model,
- Section G asks for opinions about TUIC modelling aiding non-professionals to participate within the design team.

The Patterns Questionnaire was administered at the end of the case study.

The same patterns used for Study Two were used for this study but the pattern numbering has been reorganised. Figure 8.3 shows the id-numbers better indicate the position of each pattern within the structure of the pattern language.

As a participant-observer the researcher was not a passive observer, and helped each participant by locating artefacts or information when required. The researcher with thirty years teaching experience structured the sessions to limit differences between the contexts in which each participant completed tasks. Help was provided by responding to requests for help, and taking opportunities to teach about the methods or pattern content. Some help was provided to speed up the processes by the researcher finding patterns or other artefacts on request. Field-notes for Session Two and Session Three were kept by the researcher can be found in Appendices A26 & A27.

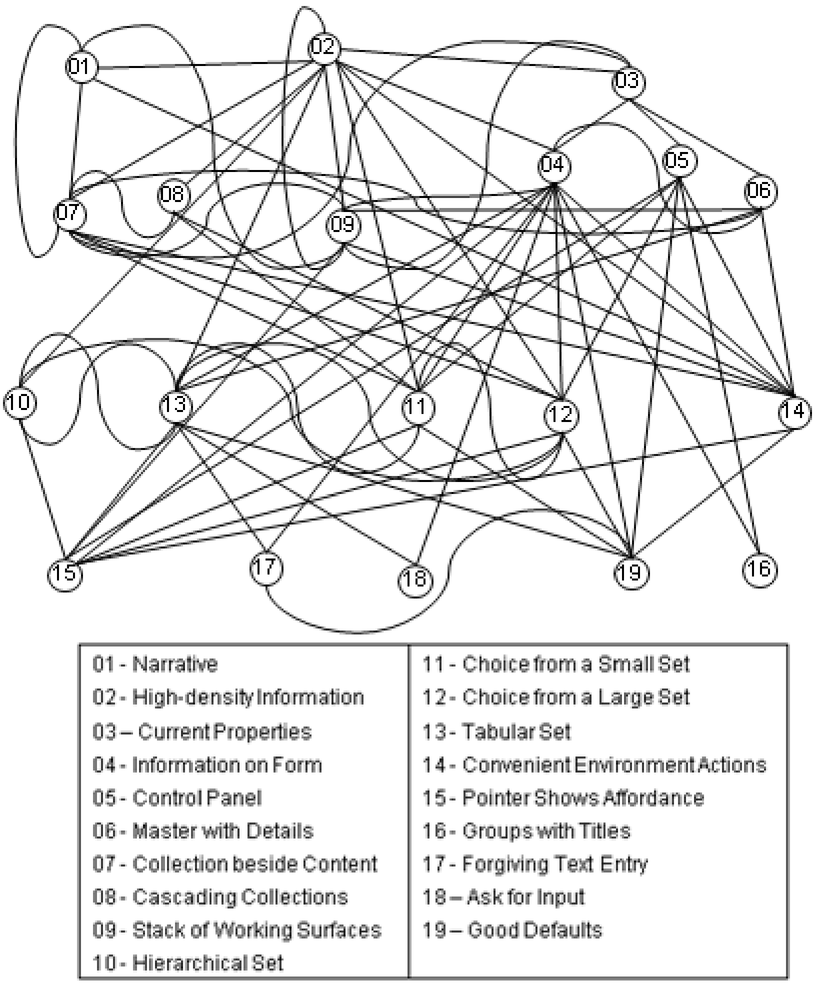


Figure 10.4 - Pattern map for the version of TUI used in Study Three

10.1.2 The Pilot Studies

Three different approaches were trialled in the pilot studies to find a suitable protocol. The pilot subjects had had many years experience as analysts and designers, including UI design. None of them had had any experience with using UI patterns or with either of the modelling techniques.

The protocol for the first pilot study was based on the experimental process used by Chung *et al.* (2004) where subjects were provided with sets of patterns to study, two days prior to a design exercise. In this pilot all the training material was provided a week prior to the proposed design session. The introductory material included: a copy of the UI patterns, the requirements documentation for the exercises and the first parts of the training manual introducing UI-pattern modelling. The subjects were asked to study this material before the first model building session. None of the potential subjects found sufficient time to study the material in depth. This approach was abandoned.

The second approach used a similar procedure to that described by Golden, John and Bass (2005). Time was provided at the start of the session for the subject to independently study all the materials, with the researcher observing. The first UI-modelling session was conducted using this approach. Even though the subject was encouraged to ask questions it was clear that studying the new material without intervention from the researcher was difficult and time consuming. The self-study option was also too time-consuming for the case studies. Trying to complete both modelling exercises in a single session as was done in Study Two also resulted in overloading the subject. The subject commented that learning two new methods and trying to design a new interface at the same time was very tiring.

The third approach involved the researcher becoming involved in the study by taking the subject through the study material and demonstrating the new modelling techniques. This approach was satisfactory and all the material was tested within the suitable timeframes.

Some rewording of the new questions in sections A, F and G of the exit questionnaire were suggested by the subjects. One subject suggested an addition to Section B to ask which sections of a pattern were actually used. Any changes made were trialled in the next pilot or reviewed with the subject who made the suggestion. One subject volunteered to edit the workbook after the modelling session.

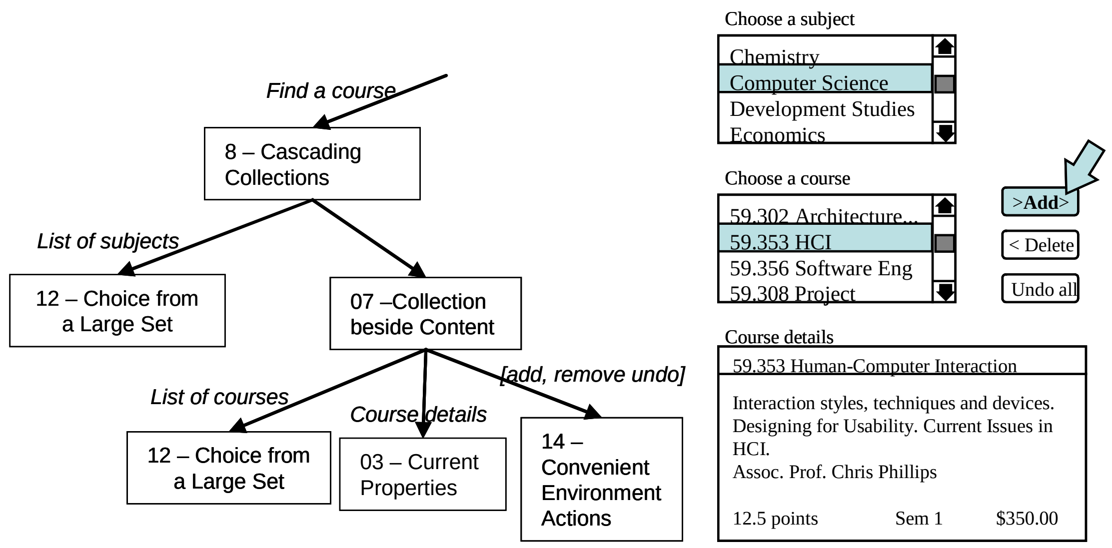


Figure 10.5 - Modified UI-pattern model showing labels on arcs beside the matching UI

A major change came about when one subject and the researcher were comparing an exemplar UI-pattern model with the exemplar UI during the final discussion. When the researcher started to annotate each component on the UI with the relevant pattern id the subject suggested adding the component's label to the corresponding arc on the UI-pattern model (Figure 10.5). These labels indicate the part of the UI being represented by that part of the UI-pattern model. Chung *et al.* (2004) reported that a pair of designers unexpectedly annotated the UI they were working on with pattern numbers and one of them commented

> *"... these [patterns] all sort of lay out the problem and the solution on a page ..." (ibid, p239)*

This matches the implicit structure of the UI with that of the selected patterns.

10.1.3 Data Collection and Analysis

The data collection approach taken was similar to that used with studies One and Two (Section 6.8). During both sessions Two and Three the researcher kept notes of actions and discussion that were pertinent to the study, therefore the subjective opinion of the researcher determined what was recorded. The researcher's observations were influenced by the types of behaviours observed in the earlier studies: such as pointing behaviours, building intermediate constructions and focussing on specific pairs of items. The researcher made notes on how each participant interacted with the questionnaires because some interactions such as looking for a specific pattern, or reviewing the exemplar UI-pattern model, were potentially important. Each within case description was validated as a correct interpretation by the participant.

The models created in each exercise were scored using a similar system to that used for the earlier studies. There are only a limited number of structures that can be created due to the limited set of UI-patterns in TUIPL. Comparison against the example solution was possible but the scoring system was flexible enough to take into account valid alternatives. These scores act as a surrogate for how well the participants understood the modelling method used.

The written responses to the sections of the exit questionnaire are reported descriptively with quotes and where appropriate interlinked with the observations. The responses to the patterns questionnaire can be summed to get an overall score for each participant's opinion of using UI patterns. The question sub-sets can also be reported using a similar technique.

The responses to the five-point scale questions in the exit questionnaire are documented in the cross-case analysis. Because there are only three participants only the raw scores and percentages will be discussed in this comparison. A simple comparison of the timings will highlight any major differences and should also identify how closely the participants followed the proposed methods. The remainder of the cross-case analysis focuses on identifying responses and opinions which were either similar across the three participants or were markedly different.

Finally, by triangulating all the threads of evidence and the subsequent discussion should confirm whether the objectives for this study have been attained.

10.2 The Case Studies

The three participants in the case studies are all currently working at a New Zealand tertiary institution. The participants have varying degrees of experience, much of it as general application developers rather than specialising in UI development. Each of the case studies follows a similar structure, reporting the participants views of UI-pattern modelling, TUIC modelling and opinions on aspects related to UI design teams under the heading of Communication.

The individual case study descriptions can be found in Appendix A24. Numeric data from the questionnaires is in Appendix A23 including model marks. Comments from the Exit Questionnaire are in Appendix A25. Stylised copies of each participants UI-pattern model is in Appendix A28 and the TUIC models in Appendix A29. Appendices A26 and A27 contain the field notes for the UI-pattern modelling and the TUIC modelling session in that order.

10.2.1 Participants' backgrounds

Characteristics	Participant One	Participant Two	Participant Three
Non-native English speaker	no	no	yes
Years of experience	6	18	10
Self rated UI experience level	past novice stage	experienced	average
Type of experience	Research projects	Full-time	Full-time
Place of experience	NZ	NZ	mostly Asia
Focus IT development	Included UI	UI	Data base
Main prototype approach	Lo-fi	High-fi	High-fi
Main UI development platforms	Delphi	Visual Studio DreamWeaver FrontPage Visual Interdev Expression Web	Delphi
Tertiary Education – HCI courses	1 undergraduate 1 graduate	1 undergraduate	None
Aware of UI patterns	Yes	Yes	No
UI-pattern modelling	Yes	Yes	No
TUIC modelling	Yes	No	No
UI modelling associated techniques	Use Case User Centered	Use Case RAD	Use Case Ad hoc

Table 10.2 - Background characteristics of participants

The background characteristics of the three participants (Table 10.2) are sufficiently varied to reduce validation threats to generalisations made from the three case studies (Flyvbjerg 2006). The participants are markedly different in a number of dimensions. The three participants have self-rated their UI design experience level very differently with one just past the novice stage, one who self-rates as average and one who self-rates as experienced. Their IT experience is six and a half years sporadic experience, ten years professional experience and eighteen years professional experience. While Participant One primarily used lo-fi prototyping as part of the UI-design process, the two more experienced developers relied on high-fi prototyping as their primary approach to UI development. All three participants had used desktop IDEs with Participant One and Three using Delphi while Participant Two had used a range of Microsoft related IDEs plus Web development tools.

All three participants have graduate level computing education but user interface design courses completed are very different. Participant One has both under and graduate experience, Participant Two has an undergraduate one while Participant Three's Computer Science qualification included no HCI courses. Both Participants One and Two had been introduced to UI patterns and UI-pattern modelling very early during this research but Participant Three had no prior knowledge of UI patterns. Participant One had also had some prior experience with CAP symbols and abstract prototyping as part of the graduate HCI design course. The only characteristic all three participants had in common was experience with the task modelling method Use Cases.

10.3 Cross-Case Comparison

The cross-case comparison brings together the findings from the three case studies comparing and contrasting the findings. The report follows a similar structure to that used for the individual case descriptions.

10.3.1 UI-Pattern Modelling

The scoring system for the UI-pattern models was similar to that used for the models created in the earlier studies (Appendix A15). The scores for the participants' UI-pattern models were all excellent.

Scoring Criteria	Participant One	Participant Two	Participant Three
Percentage of appropriate patterns	100%	100%	100%
Percentage potential patterns	92%	63%	60%
Percentage of appropriate links	95%	100%	100%
Percentage potential links	100%	100%	91%

Table 10.3 - Scores for UI-pattern models

Table 10.3 shows the scores for all criteria used to score the models. An examination of the models created shows that participants Two and Three had completed valid top levels but did not develop the lower-level details, accounting for the lower scores. The participants scored least well at identifying potential patterns (Table 10.3, line 2).

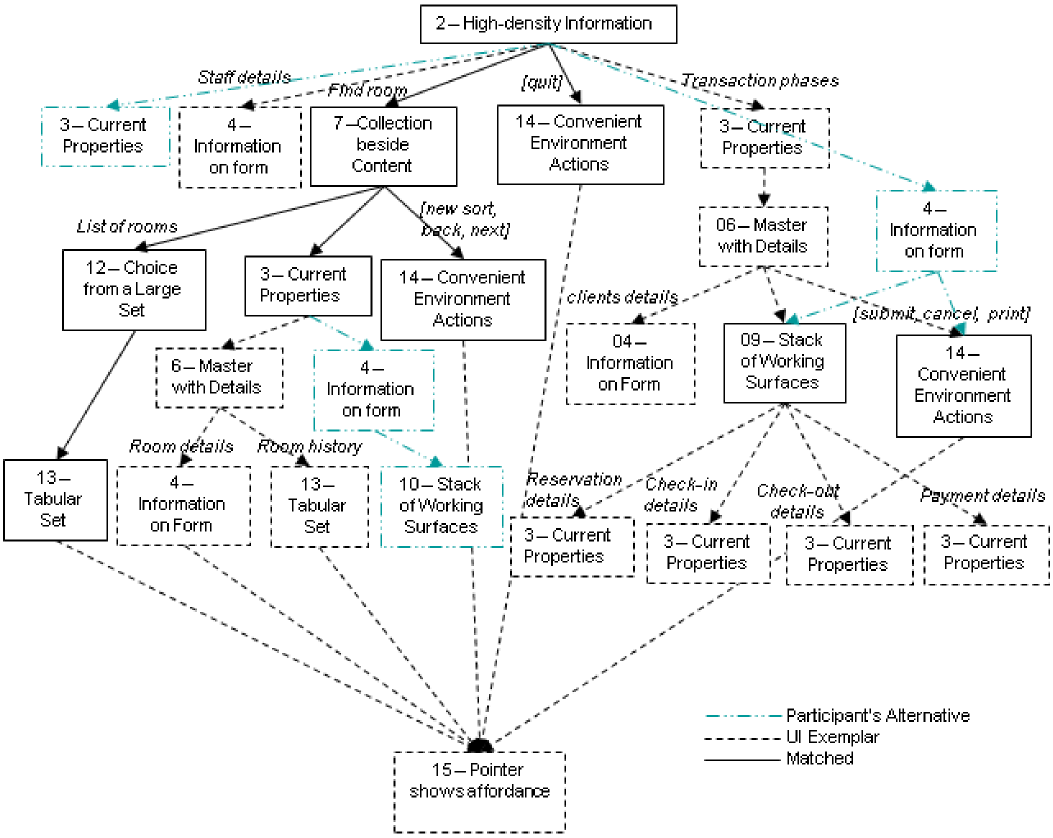


Figure 10.6 - Participant Three's UI-pattern model overlying the exemplar

This is demonstrated by Participant Three's UI-pattern model, shown in Figure 10.6. The solid lines represent those parts of Participant Three's UI-pattern model that matches the exemplar UI-pattern model. The dashed lines are the unmatched exemplar's links while the dot-dashed lines show alternative, valid links the participant selected. At the lower levels, for example pattern '09 – Stack of Working surfaces' has no reference links to the patterns it 'uses'. Participant Three chose not to model any of the four interaction areas, e.g. 'Check-in details' or 'Check-out details'.

The diagram shows that valid alternatives were chosen for representing 'Staff details' and the top level for 'Transaction phases'. Lower down they chose to use a stack of working surfaces rather than a Master detail structure for 'Room details' etc.

UI-pattern Modelling	Participant One	Participant Two	Participant Three
Intro & demo	59	33	53
Study of requirements	23	7	19
UI-pattern modelling	46	42	48
Questionnaire	15	12	23
Discussion	3	3	2
Total	2 hrs 26 mins	1 hr 37 mins	2 hrs 22 mins

Table 10.4 - Time to completed activities in the UI-pattern modelling session

The time taken to complete the UI-pattern modelling exercise was similar for all three participants, averaging forty-five minutes (Table 10.4). Participants One and Three spent a significant amount of time in discussion during the demonstration and in studying the materials prior to starting the modelling exercise. Participant Two took approximately half that time on the pre-exercise activities which may account for perceiving UI-pattern modelling as difficult. But they still produced a creditable UI-pattern model.

In discussions during the demonstrations both Participants One and Two queried how the patterns representing the lower levels of 'Finding a Course' related to the requirements. Participant Three also queried the representation of the details of this interaction space, but at a later point during the exercise. All three participants queried the role of pattern '03 Current properties' particularly in chains of linked patterns. Discussion to clarify such chains used the tutorial example to demonstrate how a higher level pattern in the UI-pattern model provided a more abstract view of the UI.

Both Participants Two and Three expressed some concern about abstract UI concepts as represented by the higher-level patterns in the UI-pattern map. The problem they had is illustrated by the difficulties they experienced with positioning compound patterns such as pattern '03 Current Properties' within their UI-pattern model. This pattern is related to a group of patterns via an 'is-a' relationship. For example pattern 03 can be completed using any one of the three patterns '04 Information on Form', '05 Control Panel' or '06 Master with Details'.

Both participants Two and Three used the examples on the back of the patterns as their primary source of information for deciding which pattern to select and for determining the intent of a pattern. These behaviours seem compatible with their expressed preference for using hi-fi prototyping when designing new user interfaces. Both identified a toolbox component that they understood in one of the example illustrations and appeared to equate that to the pattern, rather than generalise the action or presentation that the pattern represented.

Parts of a pattern	Participant
Identifier - Name	1, 2, 3
Context	1, 2, 3
Problem	3
Solution	3
(diagram)	1
Reference	1, 3
Forces	
Discussion	
Examples	1, 2, 3

Table 10.5 - Pattern sections considered most useful when completing the UI-pattern model

Table 10.5 shows that the participants viewed, some sections of the patterns as more useful than others. Neither the forces section nor the discussion section supporting the solution were identified as most useful. There was agreement among the three participants that a pattern's name, context section and the examples were all useful. Participant Two used primarily a bottom-up approach to UI-pattern modelling, which explains why the reference section wasn't considered useful. Participant Three recognised that when beginning to use UI patterns the problem and solution sections would be very important, but noted that on becoming familiar with the patterns, the name, context and reference sections would be the primary focus of attention.

Further evidence about how difficult abstraction and conceptualisation can be, came from participants' views of the TUIC model diagrams illustrating each pattern's solution. Participant One, who had prior experience with TUIC components, was very positive about the TUIC model diagrams and found them helpful when deciding whether to select a pattern. Neither Participant Two nor Participant Three thought the TUIC model diagram was particularly helpful. Both considered it took more time to comprehend the diagrams compared to just examining the examples.

Tool/Technique category	N	Participant One	Participant Two	Participant Three
UI-pattern Modelling	6	100%	67%	67%

Table 10.6 - Summary of responses to UI-pattern modelling statements

The responses to statements covering UI-pattern modelling in the Patterns Questionnaire are shown in Table 10.6. To enable comparison, the results are shown as percentages of the number of 'agree' responses to the total number of statements in the category.

Participants Two and Three recognised that they found abstraction quite difficult. This is supported by responses to three of the six statements, and supported by comments in the exit questionnaire. Participant Two did not find UI-pattern modelling particularly easy and was not confident at using the patterns to create the UI-pattern model but commented that the actual process of creating the models was straightforward. Participant Three was unsure about how easy UI-pattern modelling was and was also unsure about how straightforward the method was. This confirmed comments made in the exit questionnaire that indicated that more direction about when to repeat an activity was required. Participant One responded positively to all the statements but suggested that using colour might be helpful in describing the method.

Even with these reservations, all three participants expressed the view that developing the UI-pattern model was a useful technique in specifying what was required for a user interface. These views were confirmed by their responses to the remaining three of the six questions. All three participants agreed that they enjoyed building the UI-pattern model and that the patterns helped them focus their thoughts.

Tool/Technique category	Participant One	Participant Two	Participant Three
Developing a UI-pattern Model	Extremely helpful	Very helpful	Extremely helpful

Table 10.7 - Ratings for opinion of UI-pattern modelling

Table 10.7 shows that all three participants rated UI-pattern modelling as either extremely helpful or very helpful. Participant Two, who used a bottom-up approach to modelling, suggested that some indication of how to use the method in different modes (top-down & bottom-up) might be helpful.

Both participants One and Three considered developing a UI-pattern model would be extremely helpful in assisting non-professionals to participate more fully in a UI design team. Even Participant Two, who had had some difficulties with UI-pattern modelling, still thought UI-pattern modelling would be very helpful. Supporting comments from all three indicated that they thought that by developing the UI-pattern model non-professionals in the design team would be better able to communicate their ideas. The patterns would provide non-professionals with an appropriate technical vocabulary as well as an understanding of the UI design process. Professionals on a design team could also use the patterns to provide better explanations in support of design decisions. All three indicated that by developing a UI-pattern model they thought the final UI would better meet the needs of the users.

10.3.2 TUIC Modelling

The scoring system for the TUIC models was similar to that used in the earlier studies. When the prototypes were compared, although the positioning of the TUIC components was different, the essential elements were similar.

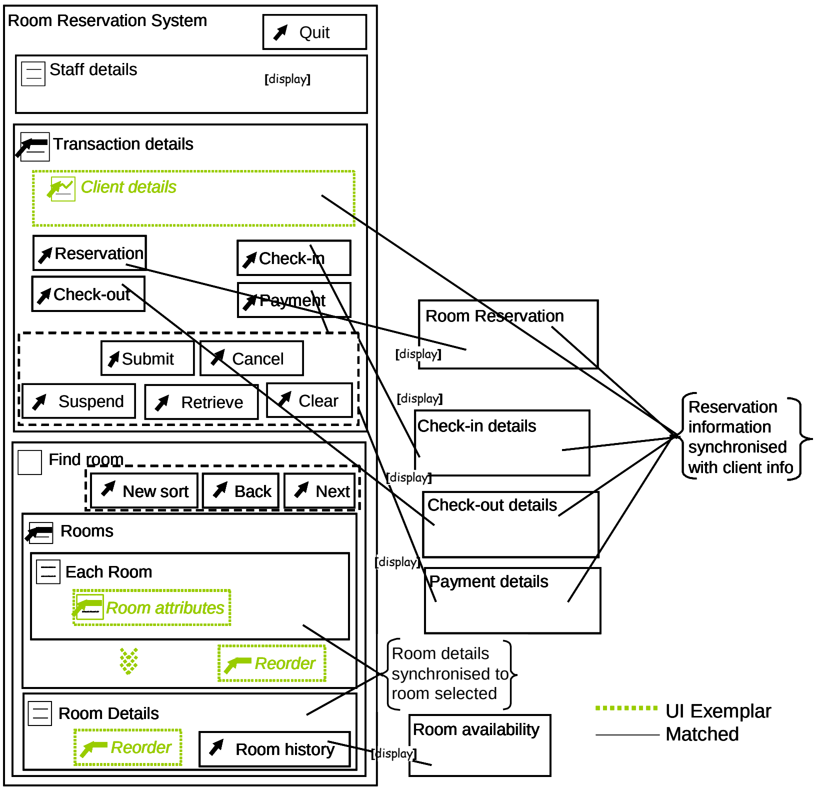


Figure 10.7 - Participant Two's TUIC model overlaying the exemplar

To score these models the components identified by the participant were overlaid on the exemplar as shown in Figure 10.7.

Scoring Criteria	Participant One	Participant Two	Participant Three
Percentage of appropriate TUIC components	100%	100%	95%
Percentage potential TUIC components	86%	90%	83%
Percentage of correct nesting	100%	100%	100%
Percentage potential labelling	100%	100%	100%

Table 10.8 - Scores for the TUIC models

The scores for the TUIC models were excellent (Table 10.8). The lower scores in line two are related to identifying all the potential TUIC components. None of the participants added a TUIC component to represent client details. Only Participant Two added some TUIC component details for the table of room data. Participant Three missed a TUIC component representing a group of actions and one synchronisation comment. Participant Three was the only person who had no prior knowledge of UI patterns and also had the least experience with UI abstract modelling, but still created a TUIC model of comparable quality to the other two.

TUIC modelling	Participant One	Participant Two	Participant Three
Intro & demo	28	18	27
Review requirements	2	7	5
TUIC modelling	18	42	26
Questionnaire	40	22	31
Discussion	5	5	4
Total	1 hr 33 mins	1 hr 34 mins	1 hr 33 mins

Table 10.9 - Time to completed activities in the TUIC modelling session

Table 10.9 shows that all three participants completed this session within one hour and thirty-five minutes but the time spent by each participant on the modelling exercise was markedly different. Both Participants One and Three spent about the same time in discussion and observation during the demonstration. As expected Participant One, who had prior knowledge of TUIC modelling, completed the modelling exercise fastest (18 minutes). Participant Three, with no prior experience, took eight minutes longer to complete the prototype (26 minutes). Participant Two also had no prior experience but had a very different time profile. The demonstration concluded quite quickly because no questions were asked. Completing the TUIC model took significantly longer. These times were commensurate with Participant Two's expressed preference for learning-by-doing.

A variety of behaviours were observed. Participant One was the only person who spent time reviewing the TUIC model by checking back to some of the patterns. Participant Two recorded progress by ticking the patterns off on the UI-pattern model as the TUIC model developed. Participant Three spread the patterns out on the work surface so they could see the diagrams of related patterns together.

Parts of a pattern	Participant
Identifier - Name	1, 2, 3
Context	3
Problem	3
Solution	1, 2, 3
(diagram)	1, 2, 3
Reference	3
Forces	
Discussion	
Examples	

Table 10.10 - Pattern sections considered most useful when completing the TUIC model

Table 10.10 shows all three participants agreed that for TUIC modelling the name, solution and diagram were most useful. Participant Three pointed out that the UI-pattern model provided all the information required about which patterns to use and how they were related, so that reference to other sections like context, reference and examples were not required.

Tool/Technique category	N	Participant One	Participant Two	Participant Three
TUIC modelling	7	86%	86%	71%

Table 10.11 - Summary of responses to TUIC modelling statements

The Three participants all agreed with four of the exit questionnaire statements that patterns helped them focus on the TUIC modelling process, that the process was relatively easy and straightforward to perform and that they all enjoyed creating the TUIC model (Table 10.11). Supporting comments in the Exit Questionnaire were very positive using words such as useful, easy, enjoyable and fun. Participant One commented that the pattern-guided method for Developing TUIC models was much easier than working directly from Use Case descriptions. Participant Two commented that by completing the TUIC model they could now appreciate the role of the UI-pattern modelling process.

Responses to the 'interpretational' diagrams varied. All three participants had minimal problems using TUIC components, although Participant Three spent some time querying parts of the symbol list near the beginning of the session. However, Participant Three was unsure how the diagrams illustrated a pattern's solution, as indicate by responses to the two related statements in the Patterns Questionnaire. All participants provided supporting comments that indicated these TUIC model diagrams made designing the prototypes relatively easy and removed ambiguity. For example Participant One commented that the diagrams helped visualize the different parts of the UI. All three participants queried aspects of how to represent a tabular list needed to represent data for available rooms.

Participants One and Two, both native English speakers, considered the description of the TUIC modelling method to be rather wordy. Participant Two suggested two versions should be created, one for those new to UI design and one for experts. Participant Three found the method useful because it clarified actions. These comments confirmed the participants' rating of the Exit Questionnaire statement about the TUIC modelling method. Both Participants One and Two were undecided but Participant Three agreed that it had the correct number of steps.

Tool/Technique category	Participant One	Participant Two	Participant Three
Patterns guide TUIC modelling	Extremely helpful	Extremely helpful	Extremely helpful
Suitability of TUIC modelling	Extremely helpful	Very helpful	Very helpful

Table 10.12 - Ratings for opinions related to TUIC modelling

The three participants all rated using the TUIC model diagrams illustrating each pattern forming the UI-pattern model as extremely helpful for creating the prototypes (Table

10.12). Participant One supported this view by commenting that using the UI-pattern model as a guide was much easier than prior experience of modelling directly from the requirements documentation. Participant Two found UI-pattern modelling initially difficult, rating it as only very helpful after the UI-pattern modelling session. But after the following session, they both expressed enthusiasm for using the UI-pattern model to guide TUIC modelling.

Participants Two and Three rated the suitability of using TUIC models for conceptual modelling in a professional UI design situation as very helpful. Participant One was of the opinion that TUIC modelling would be extremely helpful. Associated comments indicated that the process would help clarify requirements and allow solutions to be expressed clearly.

10.3.3 Communication

Tool/Technique category	Participant One	Participant Two	Participant Three
Patterns impact on communications	Extremely helpful	Extremely helpful	Helpful
UI-pattern modelling on communications	Extremely helpful	Very helpful	Extremely helpful
TUIC modelling on communications	Very helpful	Extremely helpful	Very helpful

Table 10.13 - Ratings of pattern use, modelling and communications

Table 10.13 shows that both Participants One and Two rated patterns as being extremely useful for helping non-professionals communicate in a UI design team context. They both identified that patterns should help the non-professional develop an understanding of technical terms and also that the patterns would help the team develop a common terminology and hence a common understanding of the problem. For example Participant Two said "*Technical people often use technical terms, the patterns allow unfamiliar items to non-professionals to be discussed*". Participant Three had the least UI development experience so their opinion has added weight. They rated the patterns as helpful but identified potential problems non-professionals may experience, for example some patterns names were considered to be rather abstract.

Participants One and Three both rated UI-pattern modelling as extremely helpful. Participant Two rated this method as very helpful commenting that UI-pattern modelling helped team members "*see what the proposed interface will offer and can therefore more easily and efficiently identify things that should be modified or things that are missing*".

Participant Two thought that TUIC modelling would be extremely helpful in assisting non-professionals to bridge any communication gap with other team members.

Participants One and Three rated this prototyping approach as very helpful. Participant One thought it would help team members argue their points of view more effectively. Participant One's comment also refers to the advantage of the UI patterns guiding the TUIC modelling saying "*Team members can more easily argue their point of view about modelling things in the TUIC model when they have the UI patterns solution (text and diagram) sections to help support their argument*". Participant Three thought it would help team members to organize complex problems.

10.3.4 UI Patterns

The participants' responses to statements making up the Patterns Questionnaire are considered first to identify whether there were any unexpected differences in opinions when working with UI patterns

Tool/Technique category	N	Participant One	Participant Two	Participant Three
All UI pattern statements	32	97%	72%	69%
Patterns and modelling	7	100%	57%	100%
Patterns were helpful	4	100%	100%	75%

Table 10.14 - Comparison of responses to the Patterns Questionnaire

Responses to Patterns Questionnaire statements are shown in Table 10.14. For comparison purposes the results are shown as percentages of the number of agree responses to the total number of statements in each category. The combined value for all responses to the statements is positive.

Participant One who had had the most experience with UI-patterns, responded most positively. All three Participants responded positively to four of the seven statements in the Patterns Questionnaire considering the use of patterns generally for UI modelling. Participant Two was unsure about the remaining three. These statements were concerned with whether it was easy to learn how to use the patterns in the modelling exercises, and how straight-forward it was to locate the required patterns. Participant Two initially had difficulty learning how to locate patterns when creating the UI-pattern model. The response rate of fifty-seven percent confirmed that experience. However, Participant Two recognised that UI patterns were helpful and comments in the Exit Questionnaire indicated support for UI-pattern modelling.

Four statements explicitly referred to aspects of helpfulness of patterns and both Participants One and Two responded positively to all these statements. Participant Three responded positively to three, being unsure whether the TUIC model diagram identified the essentials of a pattern's solution. These responses are supported by comments from their completed Exit Questionnaires. Participant One indicated they liked the way patterns provided suggestions for many different solutions to a problem. Participant Two liked the way patterns could be used to construct a UI design. Participant Three agreed with the other participants and also thought using UI patterns would help reduce changes later in the UI design life-cycle.

Overall the responses were positive for pattern use. Of the total ninety-six statements assessed (32 statements per participants), only four 'disagree' responses and sixteen 'undecided' responses were given.

Parts of a pattern	Participant One	Participant Two	Participant Three
Identifier - Name	UI-pattern model TUIC model	UI-pattern model TUIC model	UI-pattern model TUIC model
Context	UI-pattern model	UI-pattern model	UI-pattern model
Problem			UI-pattern model
Solution	TUIC model	TUIC model	UI-pattern model
(diagram)	UI-pattern model TUIC model	TUIC model	TUIC model
Reference	UI-pattern model		UI-pattern model
Forces			
Discussion			
Examples	UI-pattern model	UI-pattern model	UI-pattern model

Table 10.15 - Pattern sections considered most useful when completing the two models

Table 10.15 compares the importance the participants placed on the different sections of the patterns when completing UI-pattern modelling and TUIC modelling. It is clear that some sections were seen as more useful than others. Neither the forces section nor the discussion section supporting the solution, were identified as most useful for either modelling exercises. When asked, all three participants commented that they considered these sections as necessary especially when working in a team and when alternative patterns were being considered. They also viewed these sections as necessary background knowledge for non-professionals, indicating that access to such knowledge should improve communication between team members. Participant Two commented that the Discussion Section provided information the professional could use to help other team members better understand issues being discussed.

There was agreement among the three participants that a pattern's Name, Context Section, and the Examples, were all useful while pattern modelling. Participant Two used primarily a bottom-up approach to UI-pattern modelling, which explains why the Reference Section wasn't selected. Only Participant One identified the TUIC model diagram as helpful during UI-pattern modelling. This is understandable because this was the only participant who had had any experience with the TUIC component set. Participant Three recognised that when beginning to use patterns the problem and solution sections would be very important, but as a designer became familiar with the different patterns the name, context and reference sections would be the focus of attention.

10.4 Discussion – The Case Studies

The three case studies extend the exploratory research trialled in Studies One and Two. Generalisations from these case studies are recognised as potentially inconclusive, but efforts have been made to control factors that could compromise validity. For example, care has been taken in selecting three participants with varied backgrounds and experience in UI development. Given the limitations the following discussion provides some useful insights into UI-pattern use in a design situation. The discussion first considers whether the two methods could be used successfully by professional software developers to create a conceptual model for a new UI. Secondly, whether the professionals consider the methods could be used professionally.

10.4.1 Objective One- Develop Conceptual UI Models

Determine whether a UI developer can use the first two processes in TUIPL to develop conceptual UI models for a new user interface. That is:

- Create satisfactory UI-pattern models and TUIC models,
- View UI-patterns positively and as helpful guides when modelling,
- Judge the methods from TUIPL for UI-pattern modelling and TUIC modelling as acceptable.

10.4.1.1 The Resultant Models

To determine whether the participants could use the proposed pattern-guided UI design method, the quality of the resultant models created were assessed.

	Participant One	Participant Two	Participant Three
UI-pattern modelling	97%	91%	88%
TUIC modelling	96%	98%	94%

Table 10.16 - Total scores for models and prototypes (Summary Tables 10.7 & 8)

Table 10.16 shows the overall scores for the UI-pattern models and the TUIC models. Using the grade system at Massey University these models all score an A+ (Appendix A15) which clearly demonstrates that the three participants could successfully use the modelling methods to develop a conceptual UI design for a new user interface.

The breakdown of the scores for UI-pattern models reported in the cross case analysis (Table 10.3) showed that the two more experienced UI designers, and also both participants with no prior knowledge of using CAP symbols, did not develop the lower level details of their models. This may indicate that they considered they had sufficient information to proceed to begin prototyping. The structure of the exercises did not permit iterations between this process and the next. Participants had minimal time to review the models as they were built.

Analysis of the scores for the TUIC models created by the three participants showed that details were missed. Comments indicated that the participants enjoyed developing these models and with more time would be expected to have corrected omissions.

The time variations in completing the two sessions (Table 10.4 & Table 10.9) are best explained by the differences in expressed learning preferences and did not have significant impact on the quality of the models.

10.4.1.2 UI Patterns

All three participants considered UI patterns in positive terms. The pattern format was acceptable. Some sections were perceived as more useful than other sections and these perceptions were in part dependent on the design phase and the participants learning preferences. None of the participants ranked the Forces or Discussion sections highly but comments indicated they perceived a use for these sections in team design situations.

Literature discussing UI design guided by UI patterns (Dearden *et al.* 2002a, Chung *et al.* 2004, Segerstahl & Jokel 2006) reported that UI design professionals considered UI patterns to be helpful and findings from the case studies are consistent with that view. All three participants commented positively about UI patterns in both the final discussion and in response to the appropriate questions in the Exit Questionnaire. Their comments were supported by the overall responses to the Patterns Questionnaire (Table 10.14).

10.4.1.3 The Modelling Methods

The UI-pattern model and TUIC model scores both showed that the participants successfully applied the corresponding methods. The UI-pattern model was rated positively by all participants with suggestions for changes being minor. These suggestions may not have eventuated if the questionnaire had included a copy of the diagram to accompany the detailed steps (Appendix A9). The TUIC modelling method was perceived to be straightforward to follow but there was disagreement over the wording. The two native English speakers found the details of each step to be too wordy while the third participant found them helpful. A compromise suggestion was that two forms might be required, one for novice designers and one for more experienced designers.

TUIC modelling was chosen originally as the UI conceptual design method because Constantine (2003a) considered it to be easily learned and applied. This view was supported because none of the participants had any major problems with carrying out TUIC modelling. Although some details were missed in two of the participants' prototypes these omissions were not significant. Participant One who had previous experience with using TUIC components found the pattern-guided method to be easier to use than a method that progressed directly from use case descriptions to TUIC modelling. All three participants enjoyed the TUIC modelling process with Participant Two, the most experienced in UI design, being particularly enthusiastic.

10.4.2 Objective Two – Viability for Professional UI Design

Many of the open ended questions in the exit questionnaire were to determine the participants' opinions of the viability of the first two processes from TUIPL when applied in a professional setting. Their opinions indicate whether the methods are viable professionally and hence have validity in helping students use UI patterns when designing user interfaces. It is not necessary for the methods to be used in a professional setting in order to provide an authentic experience for the students (Jonassen 1999). Therefore the methods should be seen to have the potential to:

- Improve communication between UI design team members by helping them develop a common understanding of the language being used,
- Help team members to participate equally in the design,
- Help non-professional team members understand the design processes and techniques.

The opinions of the participants are limited by their current circumstances. None of the participants were currently working in an environment where they were part of a larger UI design team. Both Participants Two and Three reported that they were working closely with their 'client' using hi-fi prototyping effectively. They considered that applying the TUIPL modelling methods was not appropriate in their current projects although they were interested in accessing on-line UI patterns sets. It is possible to infer the potential viability of the trialled techniques in a professional setting from their opinions.

A number of researchers (Borchers 2001a, Chung *et al.* 2004, Dearden *et al.* 2002a, Finlay *et al.* 2002) have reported that professional UI developers considered UI patterns contributed to the design process and identified improved communication within design teams as an important contribution. The three participants also viewed patterns positively and indicated UI-patterns would be useful for providing appropriate UI knowledge to members of a UI design team. They would help improve the quality of communication between members of a UI design team. These comments confirmed findings by other researchers, for example Chung *et al.* (2004) found "*patterns to be useful in communicating design knowledge*" (p 234)

The Exit Questionnaire elicited feedback on whether UI patterns plus the two modelling methods would aid communication (Table 10.13). Participant Three was not as positive as the other two participants with regard to whether UI-patterns helped improve communication, indicating that the names of some patterns could be clearer. Participant Three's views on this point are important because English was not their native language. Koukouletsos (2008) also found pattern names were an important factor in developing an understanding of a pattern. His participants were Greek students and he noted that often short English names translate into long clumsy Greek phrases. Other researchers have identified pattern naming as important for acceptance and understanding of patterns (Chung *et al.* 2004, Kotzé *et al.* 2006, van Biljon *et al.* 2004).

All three participants indicated that both the UI-pattern modelling and the TUIC modelling would help members in a design team participate equally on the task of UI-design. Participant Two who had the most experience, considered that the methods would help ensure non-professionals on a team were better able to participate because firstly, they would be able to use the information in the patterns to communicate their ideas. Secondly, professionals would be able to use the patterns to demonstrate their design decisions. Participant Three considered the approach was worth pursuing but identified the time required to become familiar with the UI patterns could be a limiting factor. Participant One suggested that time could be saved and the resultant UI design would better meet the needs of the user group. Comments from all three participants indicated that it was not until after completing the TUIC modelling exercise that the value of creating the UI-pattern model became clear.

The research by Finlay *et al.* (2002) showed that providing a facilitator to help designers select appropriate UI patterns resulted in improved understanding of the structure of the pattern language used. In the case studies, the researcher provided support for each participant as they developed their models, but the researcher did not actively facilitate the pattern selection. The results from the three case studies suggest that it would be possible for UI professionals to learn to use the UI-pattern modelling method without active facilitation, although the learning curve would be considerably longer. The results also indicate that care would be required to ensure all members of a team had time to develop their understanding when introducing UI-pattern modelling and TUIC modelling to a design project.

All three participants commented positively on the potential of UI-pattern modelling and TUIC modelling being helpful for improving project members' understanding of the design processes and techniques. Comments were similar to those supporting their assessments of the communication questions. Participant Two reiterated the value of non-professionals being able to develop their understanding of technical terms and also the design processes. Participant Three summed up with the statement *"The patterns organize the complex situation or problems clearly into a whole as a UI-pattern model so that all team members can communicate based on it as they build the TUIC model"* This comment supports Chung *et al.*'s (2004) finding that "*patterns helped novice designers generate designs*" (p 234).

Abstraction, and particularly patterns that provided generalised descriptions for a group of patterns linked by an 'is-a' relationship caused varying degrees of difficulty for all three participants. A similar problem was identified by Koukouletsos *et al.* (2007) working with senior students comparing web usability patterns with usability guidelines. They found students had difficulty with patterns that represented two or more guidelines and suggest that the problem maybe that "*compound patterns are close to the cognitive limitation of novice designers*" (ibid, p179). Although Participants Two and Three are not novice UI designers, experience with using UI patterns was minimal and they both expressed the opinion that they found abstraction difficult.

10.4.3 Implications for Teaching

Although not an objective of this study, the researcher was a participant-observer and became actively involved in teaching the three participants. Working intensively one-to-one provided insights into potential problems that may influence student learning.

The main omission seen in the participants' UI-pattern models was lack of lower level detail in the models. This was similar in the students' models. In part, this may be due to the limited time. It also highlights that in a teaching context the need to provide time and encouragement so that students review their models for completeness and at some stage during the process remind them to consider modelling lower-level details.

One observation the researcher made is not discussed above. The decision to introduce the participants to the modified form of the UI-pattern model in which selected arcs are labelled (Figure 10.5) appears to have made UI-pattern modelling more understandable. The three case study participants created more complete models than did the subjects of the pilot studies using the unmodified form. In the case studies compared to the pilot studies the participants did not requests as much help on how to selected patterns or on how to link them into the UI-pattern model. This observation suggests that the modified UI-pattern model will have a similar impact on models created by students using the updated structure.

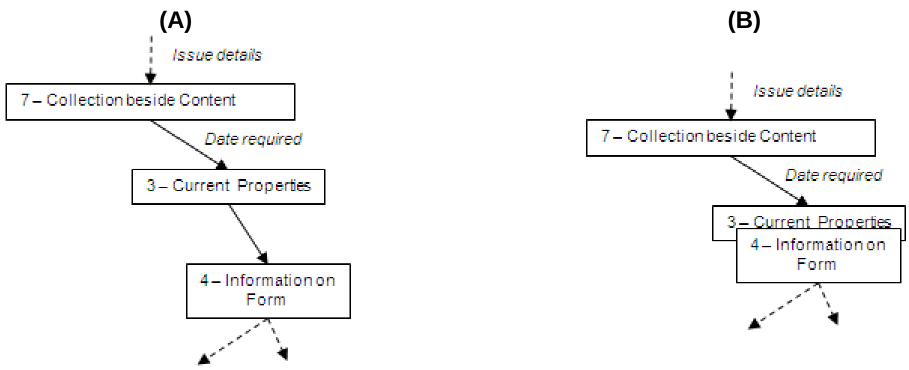


Figure 10.8 – Alternatives for representing the *is-a* relationship between patterns 3 and 4

The participants all had difficulty with the representation of the '*is-a*' relationship, as shown between patterns 3 and 4 in Figure 10.8(A). This was not identified as a specific problem by students in earlier studies but was possibly obscured by students having more general problems with UI-pattern modelling. When first introducing students to UI-pattern modelling, it may be necessary to use only the '*contains*' and '*contained-by*' relationships. Another option is to emphasize the relationship types by introducing a different notation for the '*is-a*' relationships. This would increase complexity but may improve the models clarity. A possible representation would be to remove the linking arc and overlay the symbols representing the patterns as shown in Figure 10.8(B).

Even though Participant Two had the most experience, they did not appreciate the role of the UI-pattern model until completing the TUIC model. All three participants agreed that the value of the UI pattern model became clearer after completing TUIC modelling. These responses suggest that when introducing UI-pattern modelling to students it is important to indicate how these models will then guide the development of a TUIC model. By providing an overview of the next stage of the project the student is given a clear purpose for building a complete UI-pattern model.

UI pattern naming is often identified as an issue (Wesson & Cowley 2004, Segerstahl & Jokela 2006, Saponas *et al.* 2006, Kotzé *et al.* 2006). Participant Three reported having difficulty with some pattern names. This confirms findings by Koukouletsos (2008). Students who are not native English speakers may have problems with trying to develop a common language within a team environment. These may be overcome when students are introduced to the process of instantiating the patterns to the context of the UI design when alternative names could be chosen that have more meaning for the individuals in the team.

Another issue that an educator needs to consider when introducing the different processes in TUIPL is that students have preferred learning styles (Norman 2009). This issue is clearly highlighted by the learning behaviour of Participant Two, when introduced to TUIC modelling (Table 10.9) and to a lesser extent to UI-pattern modelling (Table 10.4). Although focussing materials and teaching approach to account for learning styles appears to have little impact on outcomes (Norman 2009), allowing flexibility in the timing of activities when conducting tutorials could be important.

10.5 Discussion – Research Question Three

The three case studies were to determine whether the UI-pattern modelling and TUIC modelling methods from the TUIPL framework can provide students with a ‘real world’ UI design experience. For students, applying methods that are acceptable to professionals is important to bring authenticity to their learning experience (Jonassen 1999). The focus of the third research question is on the TUIPL framework.

Question Three: How can student UI designers be successfully guided in the use of a UI pattern language for developing conceptual UI design models?

The two UI conceptual models trialled in the three studies are the UI-pattern model and the TUIC model. The first process in TUIPL is UI-pattern modelling and the second is TUIC modelling (Figure 5.2).

Study	Objective	Observations	Models	Pattern Questions	Exit Questions
1	3 Determine whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model	Indirectly	Avg.	Count	Yes
2	2 Confirm that the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model	Indirectly	Avg.	Counts	Yes
3	1 Determine whether UI developer could apply the first two phases of the proposed pattern-guided UI design method in the development of a conceptual UI model for a new user interface	Yes	Score	Indirectly	Indirectly
3	2 Determine UI developers' opinions of the viability of the first two methods in the TUIPL framework in a professional setting				Yes
				TUIC Test	
2	3 Determine whether the TUIC modelling method from TUIPL will support students using UI patterns as a guide to the creation of TUIC models.	Indirectly	Avg.	Avg. Correlation	Yes

Table 10.17- Overview of findings from Studies One, Two and Three that support the third research question

Aspects of this question were investigated in all three Studies and the objectives from Studies One, Two and Three that directly relate to the third research question are listed in Table 10.17. It provides an overview of the findings from each of the different types of data collected. The results that indirectly support the objective have been prefaced with the word 'indirectly', simple counts, averages and a correlation have been indicated. Although the participants in Study Three were not undergraduate UI students they were required to use the proposed method while learning how to use the UI pattern language to guide their development of the two types of model. Their learning experience is important to this goal because it provides evidence of the viability of the TUIPL framework in a professional setting.

10.5.1 UI Conceptual Models

The quality of the UI-pattern models and TUIC models created act as surrogate measures of the success of the two associated methods from TUIPL. Results based on a surrogate need to be confirmed from other sources

> *"Because the surrogate is an indirect measure, there is danger that a change in the surrogate is not the same as a change in the original factor."* (Pfleeger 1994, p17)

The UI-pattern models created by the students in all three UI-pattern modelling exercises (Studies One and Two) were of acceptable quality with those from Study Two being significantly better (Table 8:13). Improvement is seen in both the compared scores and also when percentages of appropriate components are compared (Table 8:15). The students in Study Two completed the UI-pattern modelling exercise on average two minutes faster than those in Study One (Table 8.16). This may indicate that the modification improved the process in the second study. A number of researchers (Lin & Landay 2008, Wania & Atwood 2009) use completing a design task faster as a surrogate indicator of improvement. The responses by the students to statements in the Patterns Questionnaire indicated that many of them found the UI-pattern modelling activity comparatively difficult with only on average nineteen percent of them responding positively to the relevant statements in Study One (Figure 8.10). Responses in Study Two were more encouraging with thirty-five percent of students responding positively (Figure 8.10).

The UI-pattern models created by the three case study participants were also of a good standard (Table 10.3) with the main issue being lack of detail. More time for review and further iterations would probably have rectified this deficit.

The TUIC models created by the students in Study Two were on average very good with a mean score of seventy-six percent (Table 9.3) an A- grade (Appendix A15). The TUIC models created by the three case study participants were excellent all scoring an A+ (Table 10.8). In Study Three all three participants responded very positively to TUIC modelling and rated it highly and commenting that it was an enjoyable activity.

The average scores for both the UI-pattern models and the TUIC models created in all three studies demonstrate that variants of the two modelling methods can be successfully used by both students and professionals with different levels of UI knowledge to develop satisfactory conceptual UI designs.

10.5.2 The Modelling Methods

Evidence that the two methods from TUIPL can successfully guide students in the use of TUI for developing UI-pattern models and TUIC models comes from all four types of data collected.

The observations of student engagement while applying the modelling methods is one factor in assessing the success of the methods. In both studies students became fully engaged in the UI-pattern modelling exercises. Participants in Study Three had no problems concentrating on applying the modelling method either. Detailed analysis of observations of how the students used the simplified UI-pattern modelling method in Study One (Appendix A19) indicated that although there were deviations, all could successfully follow the simplified top-down approach. Students in Study Two also had no problems following a simplified version. The information gained from analysing the observations led to the refinement of the simplified UI-pattern modelling method (Figure A9.1) and the development of the full version (Figure A9.3) used in Study Three. The participants in Study Three successfully followed the full version to create UI-pattern models for a new UI.

A positive outcome from students using the simpler version of the method in studies One and Two was that it encouraged students to supplement the steps by introducing transitional models in both Study One (Table 7.17) and Study Two (Table 8.5), reversing steps and adding a review process at the end. A more complex, flexible version of the UI-pattern modelling method could be developed similar to that used in Study Three. But a simplified version is the recommended option when introducing the students to UI pattern language structure because it is known that it is difficult to learn how to use a pattern languages' structure (Finlay *et al.* 2002, Kotzé *et al.* 2006). When using the simplified version of the UI-pattern modelling method, the behaviour of students in creating transitional models and making minor modifications to the method should improve their learning experience. This behaviour is fundamental to the constructivist approach to learning (Jonassen 1999).

Students in Study One clearly found the UI-pattern modelling method helpful (Figure 7.14) but only two of the six comments specifically about the UI-pattern modelling method were positive (Section A18.2). Along with the comments about suggested changes, it is clear some of these students had difficulty following the pattern language structure rather than the method itself, as shown by the comment "*It wasn't always obvious which patterns belonged together*". Four of the five comments from the Study Two students were positive. It is clear the student who commented negatively was a non-native speaker of English because their comment indicated that they had difficulty with the vocabulary used in both the method and the patterns. This difficulty with vocabulary was also raised by the non-native English speaking participant in Study Three.

One of the Study Three participants was very positive about the UI-pattern modelling method while the other two were less positive. These two participants acknowledge they found coping with abstraction difficult.

The Exit Questionnaire comments about the UI pattern modelling method confirm the conclusions drawn from the Patterns Questionnaire's results that some students found UI-pattern modelling difficult but that the method was straight-forward to follow. The participants in Study Three also found UI-pattern modelling relatively difficult but all three rated the method highly (Table 8.7). On completion of the TUIC modelling activity they commented that they could then appreciate the value of the UI-pattern modelling process. When responses from both questionnaires are taken into account, the overall response is supportive of UI-pattern modelling.

Students in Study Two who followed up the UI-pattern modelling exercise with a TUIC modelling exercise commented positively about the TUIC modelling method. Observations showed they were fully engaged in creating their TUIC models. The participants in Study Three were also fully engaged in this activity. The students again introduced modifications to the simplified version of the method and some of these have been incorporated into the revised simplified version (Figure A9.2). These changes were used in developing the full version successfully applied by the participants of Study Three (Figure A9.4). Further evidence for the success of the TUIC modelling method comes from the students' and the participants in Study Three's acceptance of it as measured by their responses to the related sections of the exit questionnaire.

10.5.3 Efficacy of TUIPL

The overall tenor of the evidence from all the data (observations, model scores and responses to the questionnaires) is that both modelling methods were successful. The responses to the section of the Exit questionnaire asking students whether they thought the methods could be used in the design of new user interfaces by a UI design team were positive. The three professional participants from Study Three judged that the methods could also be used in a professional UI design context. All these findings indicate that although TUIPL was not designed for use in a 'real world ' situation, like the model-driven approach developed by Seffah and Gaffer (2007), TUIPL provides students with an 'authentic' UI design experience in the spirit of the constructivist theory of learning.

10.6 Summary

The quality of the models created in Study Three demonstrated that all three participants created acceptable UI-pattern models and TUIC models. They all viewed UI patterns positively. Overall, the evidence indicates that all three participants supported both methods and that together they could be part of a successful approach to UI design. These findings satisfy the first objective for this study.

Issues identified by other researchers were also apparent in this study. These include:

- The names of patterns can affect acceptance and comprehension,
- Users may need help with understanding and using patterns that refer to comparatively abstract concepts,
- When starting to use UI patterns the assistance of someone already familiar with UI patterns helps new users to overcome the initial learning curve.

The three participants were in agreement that communication between UI design team members would be enhanced because patterns would help members to develop a common understanding of the problem domain. In the process of building the UI-pattern model and TUIC models the UI patterns would help team members to participate equally with professionals. Non-professional team members should develop a better understanding of the UI design processes and techniques being used. These findings satisfy the second objective that the first two processes in TUIPL would be viable in a *professional* setting.

If the three participants are reasonably representative of UI-designers it is likely that UI-design teams could successfully create UI-pattern models and TUIC models for new UIs. The high level of agreement among the three participants provides justification for using the techniques trialled when teaching students about UI pattern-guided UI design.

Reflecting on the teaching experience highlighted some issues that may need to be addressed in future iterations of this design study.

1. Labelling arcs on the UI-pattern model may improve students' UI-pattern models.
2. The '*is-a*' relationships within a pattern language need to be clarified for students.
3. Students need time to reflect on their UI-pattern models and encouragement to add lower level details where necessary.
4. Students need to know that one purpose for building a UI-pattern model is to guide the subsequent development of a TUIC model.
5. Flexibility in timing for different activities during modelling tutorials is desirable.

By using UI-pattern modelling and TUIC modelling from TUIPL, student UI designers can be successfully guided in the use of a UI pattern language to develop conceptual UI design models of an acceptable quality.

Chapter 11: Conclusions and Further Work

The primary focus of this thesis has been on exploring tools and techniques for educating students in the use of UI pattern languages for detailed UI design. The research has been conducted within the framework of an education design study (Shavelson *et al.* 2003). Underlying the educational activities was a constructivist approach to learning (Jonassen 1999, Jonassen & Carr 2000, Jonassen *et al.* 1996). The research explored aspects of how to assist students learning '*about*' and '*through*' UI pattern languages (Griffiths & Pemberton 2004).

The initial project concept was to develop a computer aided learning environment (CAL) similar to Leopard which introduced students to OO-programming (Kemp *et al.* 2009). To be able to develop a suitable CAL tool it is first necessary to understand what students are required to learn and how to assist their learning. An important outcome of this research has been the development of the necessary understanding for successfully introducing students to UI pattern languages.

11.1 The Research Goal

The research goal for this thesis was to demonstrate that:

> *A user interface pattern language can guide student developers in the creation of conceptual user interface models.*

The three research questions raised by the research goal provided guidance for the research. The findings relate both directly and indirectly to answering each question have been discussed in depth at the end of appropriate chapters. Issues related to answering Question One about characterising UI-pattern languages are discussed at the end of Chapter Four (Section 4.5). Results related to Question Two, on requirements of a UI-pattern language are discussed at the end of Chapter Nine (Section 9.4). The third research question is directed at finding a pattern-driven UI design method suitable for use with students. Results related to this question are discussed at the end of Chapter Ten (Section 10.5). The following sections summarise these discussions and reflect more widely on relevant issues.

11.1.1 Question One: What are the characteristics of a UI-pattern language?

Seven attributes have been identified for characterising the internal structure of a pattern language. Seven tests for determining the value of each of these attributes form the basis for a UI pattern language maturity model (UMM). The first four of these tests are used to determine the internal validity status of a pattern language. The final three measure aspects of the secondary structure of a pattern language. External validity was not explored although the measures of UI pattern language maturity and complexity may relate to it in part.

UMM was used to evaluate six published UI pattern languages (Table 4.2), three of which had a maturity status of moderately mature (Figure 4.4). Maturity status is an important consideration for HCI educators because this research found in Study One, and confirmed in Study Two, that when students are introduced to a pattern language, an internally-valid UI pattern language structure is desirable.

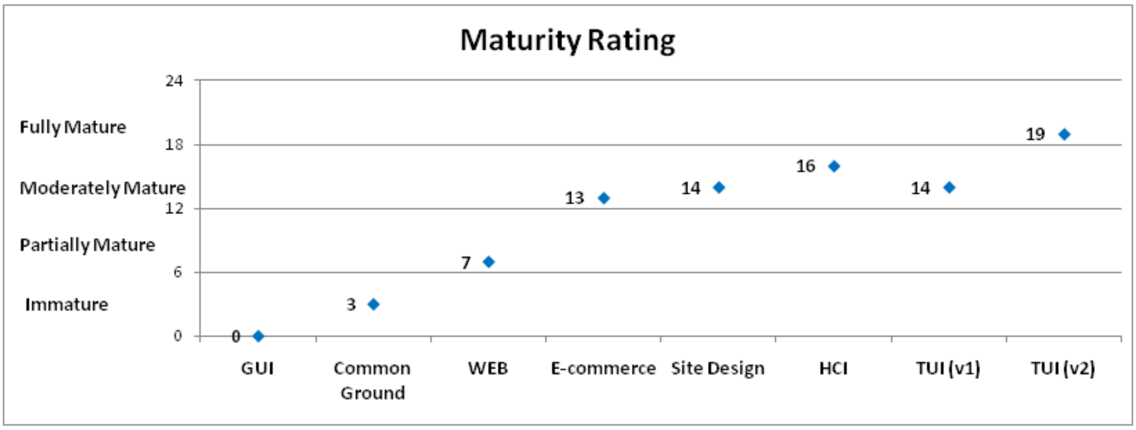


Figure 11.1 - Graph showing the maturity ranking for each of the UI pattern languages analysed including two versions of TUI

Figure 11.1 shows that the first version of TUI used for Study One was not internally valid, with a maturity rating of Moderately Mature (14). The version used for Study Two was internally valid, with a maturity rating of Fully Mature (19). Students find using an internally valid pattern language considerably easier as was demonstrated when studies One and Two were compared (Section 8.5). This suggests that the HCI educator needs to help students bridge the gap between using an internally valid version of a pattern language and a commonly available version. Educators can use the UMM to appraise the structure of existing pattern languages and identify potential problems students may encounter in applying them. Such information is particularly useful when planning modifications to help scaffold student learning (Wood & Wood 1996, van der Stuyl 2002). For example, an internally valid version of a UI pattern language could be created and then the inconsistencies found in the original structure could be re-introduced over a series of exercises. The scaffolding provided by an internally valid pattern language is thus gradually withdrawn. The students should develop the skills and confidence to be able to use the 'real world' version of the UI pattern language successfully.

The research that led to the development of the UMM also identified the need for a UI-pattern model, an enhancement of the pattern list normally used to represent a design based on UI patterns (Davis 1983). This new graphical representation clearly shows the network of selected UI-patterns describing a specific UI and explicitly identifies the required relationships between the patterns.

The educator also requires a technique for demonstrating the use of pattern language structure in UI design. The UI-pattern model appears to have promise for meeting this goal. This graphical representation illustrates and enhances the pattern list (Alexander *et al.* 1977, Davis 1983, Seffah & Forbrig 2002, Walldius 2003, Tidwell 1998, van Duyne *et al.* 2003) normally used to represent a design based on UI patterns. The UI-pattern model contains more information than the list because both the relevant relationships connecting the patterns and the levels within the hierarchy are clearly identified. The pattern list has been identified as a form of conceptual UI model (Erickson 2000, Seffah & Gaffer 2007) and therefore the UI-pattern model is also a conceptual UI model. It forms a meta-language for an example UI. Many physical UI designs can be built from the pattern descriptions identified by the UI-pattern model.

It was recognised that introducing students to the UI-pattern modelling technique has potential to scaffold student learning about pattern language structure. This premise was tested in studies One and Two. Results suggest that further research comparing the use of UI-pattern modelling with the pattern list approach is worth pursuing.

11.1.2 Question Two: Are there any specific requirements of a UI pattern language when used for teaching student UI designers?

This research has demonstrated that the structure of a UI pattern language and the form of the patterns can influence student learning when they are first introduced to them. It was found that using an internally-valid form of the TUI pattern language that was fully mature, helped to improve the models the students created. The impact of pattern illustrations on student learning was established, with statistically valid results supporting observations of other researchers (Dearden *et al.* 2002a, Chung *et al.* 2004, Sharp *et al.* 2003, Finlay *et al.* 2002). In Study One the students appeared to place undue importance on the initial illustrative example. By making these less prominent, students still had access to them and still reported that they were useful, but also appear to have paid more attention to other pattern content. The use of the interpretational diagrams in the form of TUIC models also appears to have benefitted student learning.

The patterns were rewritten so that basic relationships defining the primary structure of the pattern language are clearly distinguishable from those defining the secondary structure. The evidence indicates that this rewrite may have helped students to better understand and use the structure of the TUI pattern language.

Students have been able to follow TUIPL's methods successfully using TUI to guide the creation of conceptual UI models for existing UI examples. In studies One and Two they created creditable UI-pattern models and in Study Two creditable TUIC models. Importantly for the success of any teaching oriented pattern language, in both studies the majority of students reported positively on their experience of using TUI.

The professionals from Study Three and the students from Study Two all learned about TUIC components and TUIC modelling from working with the diagrammed UI-patterns. This was demonstrated by the scores of their TUIC models which were high. The students' TUIC test scores were also high. Being able to learn new material contained within the pattern language is also a necessary requirement of a teaching oriented pattern language.

There are still issues that need addressing. After using TUI to build the UI-pattern model, learning issues raised were similar from both the professional UI developers from Study Three and the students from studies One and Two. Both students and professionals suggesting that names of patterns might affect both comprehension and acceptance. They also saw abstract concepts present in some patterns as potential problems for both comprehension and use.

These professionals from Study Three responded positively about their experience using UI patterns, confirming findings by other researchers (Chung *et al.* 2004, Segerstahl & Jokela 2006, Lin & Landay 2008, Bernhaupt *et al.* 2009). This finding is significant because it is important to introduce students to a tool that is acceptable to professionals developers to provide the students with a realistic experience.

11.1.3 Question Three: How can student UI designers be successfully guided in the use of a UI pattern language for developing conceptual UI design models?

The discussion of this question came at the completion of Study Three and drew together the threads from all three studies. Even given all the limitations of quasi-experiments, pre-experiments and case studies, due to the weight of corroborating evidence, it is possible to conclude that the two methods from TUIPL trialled were successful, and are suitable for teaching students '*about*' and '*through*' UI patterns.

Study Three demonstrated that with a facilitator to guide learning, the professional developers created two types of UI conceptual model using the respective methods. They created creditable UI pattern models and TUIC models for a new UI based on a UI specification. All three professionals reported that they particularly enjoyed creating the TUIC models.

Study Three confirmed that using TUIPL would provide students with a UI design experience that represents real-world practice, which is fundamental to the Constructivist approach (Jonassen 1999). The results from the research also indicate that there is value in further iterations of the Design Study. For example, the response of the participants in the third study was very positive suggesting that the next stage of the Design Study should be to have students use the TUIPL framework to create conceptual models for new user interfaces.

11.2 Limitations

The three studies reported in this thesis comprise the early iterations of a design study. Design studies can be characterised by being applied in the messy situation of the real classroom (Shavelson *et al.* 2003, Collins *et al.* 2004). There are generally multiple and often complex dependent variables, and the research design is characterised by a process of flexible revision. Often variables are not controlled and interaction between participants is expected and usually encouraged. The researcher provides a description, characterising the context in which the research occurs and develops a profile of results based on both qualitative and quantitative data.

Threats to the validity of any conclusions have been identified in relevant places throughout the preceding chapters but the main limitations are reiterated here for completeness. Conclusions from design studies need to be qualified because of the recognised threats to validity of results (Brown 1992). Control groups and randomisation in the tradition of scientific experimental design are not possible. Because the research is carried out in the ‘natural’ classroom many factors may influence outcomes will be unknown and/or uncontrollable. However, multiple research methods and data collection techniques were employed, with triangulation used to provide confirmation of results and to increase confidence in any conclusions drawn from them.

The students for the two studies came from two different cohorts. They were all drawn from the population of students enrolled in the introductory HCI course over the years that this course has been taught. All these students had satisfied the same entry requirements and the same faculty had run the course for many years. In both years the same text book, lecture programme and similar assessment were used. The tutorials were presented in the equivalent week of the semester in both years. However, there is a minor risk to the validity of the results because of some unknown influence.

Care was taken to ensure the backgrounds of the three professionals who participated in the three case studies were as varied as possible to reduce threats to any conclusions drawn from them. Although the researcher, in the role of a participant-observer, facilitated each participant’s learning of the methods, care was taken to not to be pro-active in the manner of the facilitators reported in studies by Finlay *et al.* 2002. The researcher only intervened if the participant appeared to be having difficulties by inquiring if they would like some clarification. Field-notes were kept (Appendices A26 & A27).

One threat that could not be controlled was that the three professional participants from Study Three knew the researcher and this may have influenced their responses. There is a danger that they were overly sympathetic about ensuring positive outcomes for the research.

This research was exploratory and any generalisations based on the results have to be qualified. For example, if the experiments are repeated in any other teaching context the outcomes may be different. The teaching materials and suggested processes were developed then refined during the course of this research, and should be a useful resource for other researchers and educators working in this field. It is only by further iteration and replication that they will be improved and predictable outcomes in identifiable contexts will be possible. This process of successive refinement underpins the design study approach. The conclusions that follow should be read with these limitations in mind.

11.3 Research Contributions

Within the limitations of conducting research in a classroom setting, the students involved in these studies have clearly been able to use a UI pattern language to guide the creation of UI conceptual models for existing UI examples. The research shows that by creating a UI pattern language specifically structured for teaching, students can more easily use the patterns in the UI design process. The students could readily follow the two methods from TUIPL trialled in Study One and Study Two. Students created creditable UI-pattern models. In Study Two students also created creditable TUIC models. Both these models are forms of UI conceptual models.

Further support for satisfying the research goal comes from the questionnaires. In both studies the majority of students reported positively on the experience of using TUI, and the TUIPL framework. Study Three showed that with the aid of a facilitator, professional developers could develop sufficient competency with the two conceptual modelling methods in a limited time period to create creditable UI pattern models and TUIC models for a new UI based on the specification. The TUIPL framework was shown to provide the students with an 'authentic' real world experience.

The cumulative effect from the four different sources of data, results in positive outcomes, which add to the validity of the conclusion that the research goal has been satisfied.

Griffiths and Pemberton (2004) identified three ways in which UI patterns could be used to educate students: *about* UI pattern languages, *through* UI pattern languages and *with* UI pattern creation. They report on using patterns in all three ways, as did Borchers (2001a, 2002) and Barfield *et al.* (1994).

This research has explored aspects of how to assist students learning *about* and *through* UI pattern languages. The research outcomes are:

1. The development of the TUIPL framework: This is a UI pattern-guided design framework created specifically to introduce students to UI conceptual design. It was developed to provide students with a 'real world' UI design experience in line with constructivist learning theory.

2. The development of the UI-pattern model: This is a new graphical representation that enhances the sequence of patterns selected to represent a UI. It has been shown to aid in the incremental development of a UI pattern language because it identifies both patterns and relationships between patterns that require modification, addition or deletion. The process of developing the UI-pattern model can help identify where new patterns are required. This is a form of generative process identified by Alexander (1979).

3. The development of the TUIC model: This is a new form of UI conceptual model that combines the CAP symbols and navigation symbols of abstract prototyping.

4. The development of UMM: This is the first stage of a proposed UI pattern maturity model that can be used to characterise the structure of a UI pattern language and to compare existing UI pattern languages. The first four criteria tested by UMM determine whether a UI pattern language has internal validity. The other three assess the secondary structures.

5. The development of the validated pattern map: This diagram is a restricted form of the pattern map and models the primary structures characterising a pattern language.

6. The development of TUI: This is a UI pattern language specifically designed for introducing students to UI pattern languages. It is internal valid and each pattern includes a TUIC model diagram illustrating the essential elements of the solution. Illustrated examples are placed at the end of the pattern where they are accessible but not easily equated as ‘being’ the pattern.

7. A set of teaching materials: These provide artefacts and procedures for introducing students to pattern-guided UI conceptual design.

This research examined a number of foundation issues that needed to be clarified to improve the learning experiences of students when introducing them to using UI patterns in UI design. The main research findings are:

1. That UMM can be used to compare UI pattern languages.

2. That UI-pattern modelling is a suitable method for guiding the incremental development of a UI pattern language’s structure; a form of a generative process.

3. That UI-pattern modelling can be used by educators to modify UI pattern languages so they can better meet the needs of their students.

4. That by building UI pattern models educators can identify inconsistencies in existing UI pattern languages and use this knowledge to scaffold student learning.
5. That a student can use the TUI pattern language to guide them in developing two types of UI conceptual model: UI-pattern models and TUIC models.
6. That a student can use the TUIPL framework with the TUI pattern language to learn about the CAP symbols of the TUIC components and learn how to use the TUIC components to create creditable TUIC models.
7. That tailoring the form of the participating patterns and the structure of the UI pattern language, TUI enhanced learning when students were being introduced to using UI patterns in a UI design context.
8. That the position, size and type of illustration used in patterns can influence student learning outcomes.
9. That a student can develop their understanding of UI pattern language structure through the successful application of the UI-pattern modelling method.

11.4 Recommendations for Teaching

Although not directly related to the research questions, the incremental development of the instruction over the first two studies was discussed in Section 8.7. Such improvement is a fundamental goal of the over-arching Design Study. The results from this research indicate that student learning can be improved when they are introduced to UI pattern languages.

Reflections of the researcher on experiences in facilitating the learning of the participants in the case studies in Study Three, also provides insights into potential improvements that may help student learning. These were reported in Section 10.4.3.

When introducing students to UI pattern languages and UI conceptual modelling as part of an introductory HCI design course, it is recommended that educators:

1. Use a validated pattern language akin to the diagrammed version of TUI. This includes ensuring that only basic relationships are identified in the context and reference sections of the pattern. Other relationships should be introduced and discussed in another part of the pattern e.g. the discussion section of each pattern.
2. Use UI-pattern models to help convert an existing pattern language into an internally valid pattern language for use in teaching.

3. Use UI-pattern modelling to help students understand pattern language structure.
4. When modifying patterns for student use, illustrate the essential elements of the solution using interpretational illustrations such as the TUIC model diagrams.
5. Provide illustrations of archetypical examples in all patterns, but move these to a less prominent position than the beginning, near the end of the pattern form. Also ensure that size does not draw undue attention to any illustrations.
6. Introduce the students to the methods defined by TUIPL to guide their UI design processes.
7. Encourage the students to work with the UI patterns in small design groups.
8. Provide time and encouragement so students review their models, and at some stage during the process remind students to consider modelling lower level details.
9. Provide an overview of TUIPL before introducing each method so the student is given a clear purpose for building the UI-pattern model.

11.5 Future Research

This research commenced with an investigation into the nature and structure of UI pattern languages because educators need to understand what they are teaching. This led to the development of UMM, which is useful for educators to employ when selecting a suitable UI pattern language for students and for comparing UI pattern languages. When developing a teaching-oriented pattern language UMM can be used to evaluate whether the structure of the pattern language is improving. Research required to progress the development of UMM should include:

1. The application of UMM to a number of substantial UI pattern languages, after careful analysis of the relationships identified as links between patterns.
2. The development of UMM to identify key factors that characterise each stage of the maturity of a UI pattern language.
3. The use of UI-pattern modelling to model a series of UIs, and track the impact such modelling has on the development of an existing UI pattern language and its changing maturity status. Such a study could confirm the premise that modifying a UI pattern language, based on inconsistencies identified through repeated use of UI-pattern modelling, is a generative process.

Further research should investigate a number of issues raised during the iterations of the design study and progress it to the next stage. It should be able to gather evidence supporting or refuting findings from the current research. The issues are:

1. UI-pattern models were used to convert a set of existing patterns into a validated pattern language. This process could be incrementally reversed to remove scaffolding provided by the validated pattern language until students can successfully use the original 'real world' pattern language structure. A study is required to establish whether this process works.

2. UI-pattern models were introduced to help students more easily understand pattern language structure. The results of the experiments reported here indicate that there is value in carrying out further studies to investigate the creation of the UI-pattern model. Firstly, to determine whether UI-pattern modelling helps students better understand pattern languages and secondly whether students can better use the structure of a pattern language, when compared to using the standard pattern list approach. The modification of the notation to use a different representation of the 'is-a' relationship also requires testing.

3. Although not fully explored in this study the success of the TUIC model illustrations in helping students identify the essential elements of a pattern's solution indicates that using interpretational illustrations might assist students in understanding and using pattern language structure. A study needs to be designed to test whether additional diagrams illustrating the context and reference sections in a pattern can improve student understanding of using pattern language structure. For example a simple diagram could be added to the reference section of each pattern. These could be used when students are first introduced to a pattern language.

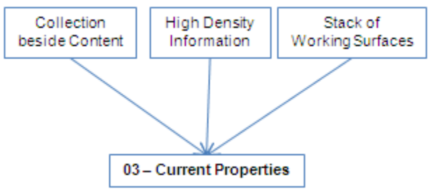
TEXT	DIAGRAM
The information describing an artefact must be displayed. The information may be changed over time. Like most patterns that can an overview description of a user interface, this pattern can also be used for the content part of the pattern COLLECTION BESIDE CONTENT or for information on a surface in a STACK OF WORKING SURFACES and also as part of a display of HIGH-DENSITY INFORMATION.	

Table 11.1 – Example of diagram that could illustrate the context sections of pattern '03 – Current Properties'

A similar diagram could be used to illustrate the context section as shown in Table 9.3. This scaffolding could be withdrawn once students have gained experience with using the pattern language.

4. The use of TUIC models as interpretational illustrations of the essential elements of a pattern's solution appears to have a positive outcome on student learning. Some patterns use annotated images of example UIs to highlight points relevant to the pattern's solution. This technique needs to be investigated to establish the extent to which it helps student learning. In a teaching oriented UI pattern language both the above techniques should probably be used together. But further studies are needed to confirm whether this would be beneficial.
5. The format of the UI patterns used in this research followed closely the form used in Tidwell's 'Common Ground' pattern language. Most recent versions of UI pattern languages have used a variant of the 'Use when' form seen in the latter versions of van Welie's 'Patterns in Interaction Design' pattern language. Further studies are needed to discover whether there is any significant difference in student perception of using the two pattern forms, and the quality of UI-pattern models and TUIC models created by students when using them.
6. Some of the UI pattern collections (van Duyne *et al.* 2003, Tidwell 1998) included questions for selecting patterns when solving a specific problem. Alexander *et al.* (1977) also includes a set of questions to guide pattern selection from different levels within the hierarchy. A set of questions guiding the selection of an initial set of patterns which are then organised into a UI-pattern map is a viable alternative methodology. Such a method should also be investigated to discover whether such a set of questions could help students make better choices in selecting appropriate patterns when working with a large UI-pattern language.
7. Only stages One and Two of the TUIPL framework have been trialled with students. A research study needs to be designed to adequately trial the third stage, where the selected patterns are modified to reflect the context of the UI being modelled.
8. A further study is required to trial the TUIPL framework with student participants creating designs for new UIs. The students should be first introduced to the methods of the different stages of TUIPL using a process similar to that described for Study Two of this research. Next TUIPL should be used to guide the creation of models from a UI specification like that first used in Study Three.

9. The aim of the overall Design Study is the specification of a computer-aided learning environment to provide educators with a tool for creating suitable learning materials and a tutoring system to guide students as they learn about UI pattern language-guided design.

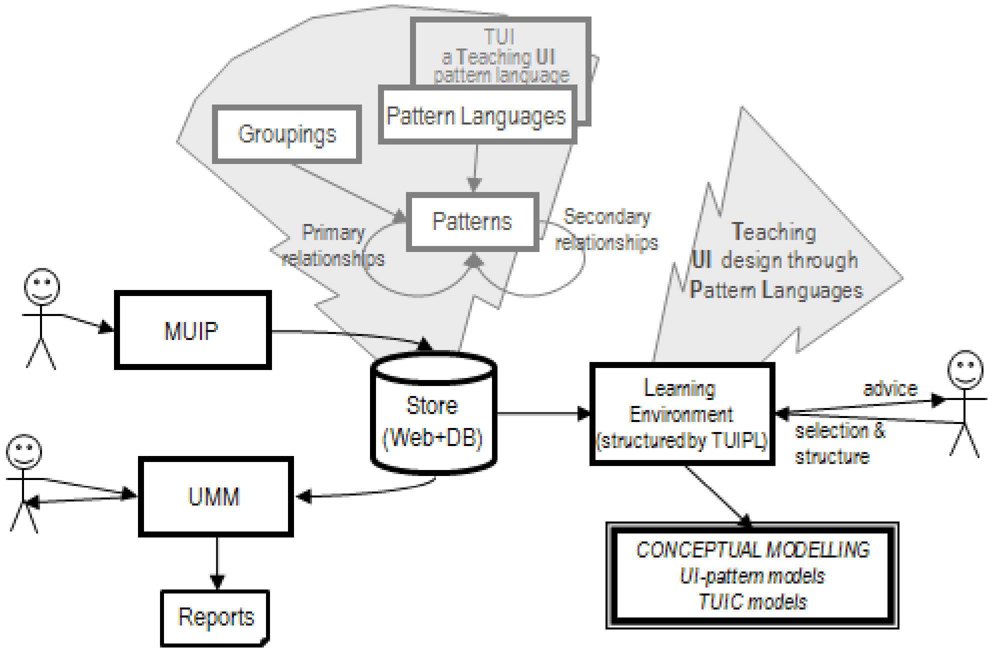


Figure 11.2 - Proposed architecture for a computer aided learning environment to teach about UI design through pattern languages

Figure 11.2 shows an overview of the architecture of such a system. This system requires a pattern management system similar to MUIP (Section 2.3.3.2), to store appropriate UI pattern languages. It should have facilities to create special-purpose teaching pattern languages, such as TUI, and enable output of stored patterns in different forms. The tutoring system would be structured around a suitable teaching-oriented pattern-guided UI design system like TUIPL, with sets of graded exercises and UI design problems based on constructivist learning theory. Ideally this system should also include visualisation tools for viewing pattern maps and validated pattern maps. Visualisation tools are also required for creating the conceptual models: UI-pattern models and TUIC models. There should also be an associated tool for reporting on the maturity of a pattern language to aid educators in the selection and development of UI pattern languages. The computer-aided learning environment should also be capable of presenting intermediate forms of a pattern language as scaffolding is withdrawn.

Abbreviations

CAP – Canonical Abstract Prototype

GUI – Graphical User Interface

HCI – Human to Computer Interface

PLML – Pattern Language Markup Language (Pel-mel)

PBL – Problem Based Learning

SE - Software Engineering

TUI - the Teaching User Interface pattern language

TUIC - Teaching User Interface Conceptual model

TUIPL – Teaching oriented User Interface design guided by a Pattern Language

UCD – User-Centered Design

UI – User Interface

UMM – UI-pattern language Maturity Model

References

A

Acosta, A. and Zambrano, N., 2004. Patterns and Objects for User Interface Construction. Journal of Object Technology, 3(3), 75-90

Adelman, L., 1991. Experiments, Quasi-Experiments, and Case Studies: A Review of Empirical Methods for Evaluating Decision Support Systems. IEEE Transactions on Systems, Man, and Cybernetics, 21(2), 293-301

Alexander, C., 1979. The Timeless Way of Building. New York, USA: Oxford University Press

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-KinI, G. and Angel, S., 1977. A Pattern Language. New York: Oxford University Press

Alexander, C., Silverstein, M., Angel, S., Ishikawa, S. and Abrams, D., 1980. The Oregon Experiment. New York: Oxford University Press

Alur, D., Crupi, J. and Malks, D., 2001. Core J2EE Patterns: Best practices and design strategies. Upper Saddle River, Sun Microsystems Press, Prentice-Hall

Alvarez, M., Conroy B. and Catena A., 2006. Criteria of the Peer-review Process for Publication of Experimental and Quasi-experimental Research in Psychology. International Journal of Clinical and Health Psychology, 6(3), 773-787

Appleton, B., 1997. Patterns and Software: Essential Concepts and Terminology. Object Magazine Online, 3(5)

B

Barfield, L., W. van Burgsteden, Lanfermeijer, R., Mulder, B., Ossewold, J., Rijken, D. and Wegner, P., 1994. Interaction design at the Utrecht School of the Arts. SIGCHI Bulletin 26(3), 49-86

Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, D., Gross, B., Lehder, D., Marmolin, H., Moore, B., Potts, C., Skousen, G. and Thomas, J., 1998. Putting it all together: Towards a pattern language for Interaction Design. SIGCHI Bulletin. 30(1), 17-24

Beck, K., and Cunningham, W., 1987. Using Pattern Languages for Object-Oriented Programs Report No. CR-87-43, Computer Research Laboratory, Tektronix, Inc

Bell, P., 2004. On the Theoretical Breadth of Design-Based Research in Education Educational Psychologist, 39(4), 243–253

Benbasat, I., Goldstein, D., and Mead, M., 1987. The case research strategy in studies of information systems. Management Information Systems Quarterly, September 368-386

Bernhaupt, R., Winckler, M. and Pontico, F., 2009. Are User Interface Pattern Languages Usable? A Report from the Trenches. In T.G.e. al (Ed.), Human-Computer Interaction – INTERACT 2009, Uppsala, Sweden. © IFIP International Federation for Information Processing, Part II. 5727/2009, 542-545

Beyer, H. and Holtzblatt, K., 2002. Contextual Design: Defining Customer Centered Systems. San Francisco: Morgan Kaufmann

Bienhaus, D., 2004. PLMLx Doc. Available at <http://www.cs.kent.ac.uk/people/staff/saf/patterns/diethelm/plmlx_doc/index.html> accessed July 2005

Bonwell, C. and Eison, J., 1991. Active Learning: Creating Excitement in the Classroom. ASHER-ERIC Higher Education Report No. 1, George Washington University, Washington, DC

Borchers, J., 2000. Breaking the Interdisciplinary Limits of Computer-human Interaction Design: A Pattern Approach. SIGCHI Bulletin, 32:1

Borchers, J., 2001a. A Pattern Approach to Interaction Design. Chichester, England: Wiley

Borchers, J., 2001b. A pattern approach to interaction design. AI and Society, 15, 359-376

Borchers, J., 2002. Teaching HCI Design Patterns: Experience From Two University Courses. Proceedings of the Conference on Human Factors in Computing Systems: Changing the World, Changing Ourselves-Patterns in Practice workshop, Minneapolis, Minnesota, USA. ACM

Brown, A., 1992. Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. Journal of the Learning Sciences, 2(2), 141-178

Bruckman, A., 1998. Community Support for Constructionist Learning. Computer Supported Cooperative Work: Journal of Collaborative Computing, 7, 47–86

Burnstein, I, Suwannasart, T., Carlson, C., 1996. Developing a Testing Maturity Model: Part I, Crosstalk: The Journal of Defense Software Engineering, August 1-7

C

Campos, P. and Nunes N., 2004. Canonsketch: a User-Centered Tool for Canonical Abstract Prototyping. Proceedings of the 11th International Workshop on Design, Specification and Verification of Interactive Systems. Springer-Verlag.

Carney, R. and Levin, J., 2002. Pictorial Illustrations Still Improve Students Learning From Text. Educational Psychology Review, 14(1), 5-26

Carroll, J., 2002. Patterns as a framework for HCI knowledge. Proceedings of the Conference on Human Factors in Computing Systems: Changing the World, Changing Ourselves-Patterns in Practice workshop, Minneapolis, Minnesota, USA. ACM

Chung, E., Hong, J., Lin, J., Prabaker, M., Landay, J. and Liu, A., n.d. Ubicomp Design Patterns. Available at <http://guir.berkeley.edu/projects/patterns/> accessed July 2007

Chung, E., Hong, J., Lin, J., Prabaker, M., Landay, J. and Liu, A.L., 2004. Development and evaluation of emerging design patterns for ubiquitous computing. In Proceedings of the 2004 Conference on Designing Interactive Systems: processes, practices, methods, and techniques, Cambridge, Massachusetts, USA, 223-242

Clancy, M. and Linn, M., 1999. Patterns and pedagogy. SIG Computer Science Education: ACM Press New York, 31, 37-42

Cobb, P., Confrey, J., di Sessa, A., Lehrer, R. and Schauble, L., 2003. Design Experiments in Educational Research. Educational Researcher, 32(1), 9-13

Collins, A., Joseph, D. and Bielaczyc, K., 2004. Design Research: Theoretical and Methodological Issues. The Journal of the Learning Sciences, 13(1). 15-42

Constantine, L., 1998. Rapid Abstract Prototyping. Software Development. Available at <http://www.foruse.com/articles/canonical.htm> accessed October 1998

Constantine, L., 2003a. Canonical abstract prototypes for abstract visual and interaction design. Proceedings, 10th International Workshop, DSV-IS 2003: Interactive Systems: Design, Specification, and Verification, Funchal, Madeira Island, Portugal, 11-13 June. Lecture Notes in Computer Science, Springer-Verlag, 2844

Constantine, L., 2003b. Usage-Centered Patterns: A Collection of Abstract Design Patterns for Presentation and Interaction Design. Constantine and Lockwood, Ltd. Available at <http://www.foruse.com/articles/abstract.htm> accessed June 2006

Constantine, L., Windl, H., Noble, J. and Lockwood, L., 2003. From Abstraction to Realization: Canonical Abstract Prototypes for User Interface Design. Constantine and Lockwood Ltd. Available at <http://www.foruse.com/articles/abstract.htm> accessed June 2006

Cook, N., 1994. Varieties of knowledge elicitation techniques. International Journal of Human-Computer Studies, 41(6), 801-849

Coplien, J. and Schmidt, D., (Eds.) 1995. Pattern Languages of Program Design. Reading, Addison-Wesley

Cowley, N., and Wesson, J., 2005. An experiment to measure the usefulness of patterns in the interaction design process. Lecture notes in computer science, Springer, 3585, 1142-1145

Crabtree, A., Hemmings, T. and Rodden, T., 2002. Pattern-based support for interactive design in domestic settings. Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, London, England. ACM Press, 265-276

Cybulski, J. and Linden, T., 2000. Teaching systems analysis and design using multimedia and patterns. Proceedings of the 13th Conference on Software Engineering Education and Training

D

Davis, H., 1983. Individual Houses in Groups: The Pattern Language in a Teaching Studio. Journal of Architectural Education-JAE, 36(3), 14-19 de Paula, M. and Barbosa, S., 2003. Bringing Interaction Specifications to HCI Design Patterns. Proceedings of New Horizons: Perspectives on HCI Patterns: Concepts and Tools workshop, Fort Lauderdale, Florida, USA. ACM

Dearden, A. and Finlay, J., 2006. Pattern Languages in HCI: A Critical Review. Human-Computer Interaction, 21(1), 49-102

Dearden, A., Finlay, J., Allgar, E. and McManus, B., 2002a. Evaluating pattern languages in participatory design. Proceedings of the Conference on Human Factors in Computing Systems: Changing the World, Changing Ourselves, Minneapolis, Minnesota, USA. ACM, 664-665

Dearden, A., Finlay, J., Allgar, E. and McManus, B., 2002b. Using Pattern Languages in Participatory Design. Proceedings of the Participatory Design Conference, Palo Alto, CA. CPSR, 104-113

Deng, J., Kemp, E. and Todd, E., 2005. Managing UI Pattern Collections. Proceedings of 6th Annual Conference of the NZ ACM SIGCHI: Making CHI Natural: Auckland, New Zealand

Deng, J., Kemp, E. and Todd, E., 2006. Focussing on a standard pattern form: The development and evaluation of MUIP. Proceedings of 7th Annual Conference of the NZ ACM SIGCHI: Design Centered HCI, University of Canterbury, Christchurch, New Zealand. ACM press

Dovey, K., 1990. The pattern language and its enemies. Design Studies. 11(1) January, 3-9

E

Erickson, 2000. Lingua francas for Design: Sacred Place and Pattern Languages. Proceedings of the conference on Designing Interactive Systems, Brooklyn, NY, 357-368

F

Fincher, S., 1999. What is a Pattern Language? Patterns workshop at *INTERACT'99*, Edinburgh. Available at <http://www.cs.ukc.ac.uk/people/staff/saf/patterns/> accessed February 2003

Fincher, S., 2003a. Perspectives on HCI Patterns: Concepts and tools. Interfaces, 56, 26-28

Fincher, S., 2003b. Some thoughts on Structuring Principles. Proceedings of New Horizons workshop: Perspectives on HCI Patterns: Concepts and Tools, Fort Lauderdale, Florida, USA

Fincher, S. and Utting, I., 2002. Pedagogical Patterns: their Place in the Genre, The 7th Annual Conference on Innovation and Technology in Computer Science Education

Finlay, J. Allgar E., Dearden, A. and McManus, B., 2002. Pattern languages in participatory design. Proceedings of HCI, People and Computers XVI—Memorable yet Invisible, 159–174

Finlayson, C., 2005. Coastal Trading. NZ House and Garden. March, 36-43

Flyvbjerg, B., 2006. Five Misunderstandings About Case-Study Research. Qualitative Inquiry, 12(2), 219-245

Folmer, E., van Welie, M. and Bosch, J., 2006. Bridging patterns: An approach to bridge gaps between SE and HCI. Information and Software Technology, 48, 69–89

Frizell, S. and Hübscher, R., 2002. Aligning Theory and Web-based Instructional Design Practice with Design Patterns. Proceedings of World conference on E-Learning in Corporate, Government, Healthcare, and Higher learning, Montreal, Canada. AACE

Freeman, E., Freeman, E., Bates, B., and Sierra, K., 2004. Head First Design Patterns. O' Reilly & Associates, Inc

G

Gaffar, A., Sinnig, D., Javahery, H. and Seffah, A., 2003. MOUDIL: A Comprehensive Framework for Disseminating and Sharing HCI Patterns. Proceedings of New Horizons workshop: Perspectives on HCI Patterns: Concepts and Tools, Fort Lauderdale, Florida, USA

Galliers, R. D., 1992. Choosing Information Systems Research Approaches. Information Systems Research: Issues, methods, and practical guidelines. R. D. Galliers. Oxford, Blackwell: 144-162

Galliers, R., and Land, F., 1987. Viewpoint: choosing Appropriate Information Systems Research Methodologies. Communications of the ACM, 30(11), 900-902

Gamma, E., Helm, R., Johnson, R. and Vlissides, J., 1995. Design patterns: Elements of reusable object orientated software. Reading, MA, Addison Wesley Longman

Garzotto, F., Retali, S., Tzanavari, A. and Cantoni, I., 2004. From Pedagogical Paradigms to Hypermedia Design Patterns: Where to Start? Proceedings of the 2004 World Conference on Educational Multimedia, Hypermedia and Telecommunications, Lugano, Switzerland

Gerring, J., 2004. What Is a Case Study and What Is It Good for? American Political Science Review, 98(2), 341-354

Golden, E., John, B.E., and Bass, L., 2005. Quality vs. quantity: comparing evaluation methods in a usability-focused software architecture modification task, International Symposium on Empirical Software Engineering, IEEE, 10.

Goldfedder, B. and Rising, L., 1996. A Training Experience with Patterns. Communications of the ACM, 39(10), 60-64

Graham, I., 2003. A pattern language for web usability: Addison-Wesley

Graham, I., n.d. The "Wu" Pattern Collection Web site. Available at <http://www.trireme.com/WU/whatis.htm> accessed July 2007

Granlund, A., Lafreniere, D. and Carr, D., 2001. A Pattern-Supported Approach to the User Interface Design Process. Proceedings of the Conference on Human Factors in Computing Systems: Anyone. Anywhere. New Orleans, USA

Greene, S.L., Matchen, P.M. and Jones, L., 2002. Position Paper. Proceedings of CHI 2002: Patterns in Practice workshop, Minneapolis, Minnesota, USA. ACM

Greene, S.L., Matchen, P.M., Jones, L., Thomas, J.C. and Callery, M., 2003. Tool-based Decision Support for Pattern Assisted Development. Proceedings of CHI 2003: Perspectives on HCI Patterns workshop:, Fort Lauderdale, Florida, USA. ACM, 5, 15

Griffiths, R. and Pemberton, L., 2001. Don’t Write Guidelines Write Patterns! Available at <http://www.it.bton.ac.uk/staff/lp22/guidelinesdraft.html> accessed Dec 2002

Griffiths, R.N. and Pemberton, L., 2004. Teaching Usability Design Through Pattern Language. Available at <http://www.it.bton.ac.uk/staff/lp22/CHIpaper.html> accessed July 2007

Guy, E.S., 2003. Developing a pattern language from observation of the development of shared information spaces. Proceedings of the Eighth European Conference on Computer Supported Cooperative Work-Workshop: From Good Practices to Patterns, Helsinki, Finland

H

Heffne, C., 2004. Pre-experimental Design from Research Methods. Available at <http://allpsych.com/researchmethods/researchcontents.html> accessed June 2009

Henninger, S., 2007. A Framework for Flexible and Executable Usability Patterns Standards, In The 31st Annual Software Engineering Workshop. Baltimore, USA

Henninger, S. and Corrêa, V., 2007. Software Pattern Communities: Current Practices and Challenges. The 14th Conference on Pattern Languages of Programs (PLoP 07). Monticello, Illinois, USA

Henninger, S., Keshk, M. and Kinworthy, R., 2003. Capturing and Disseminating Usability Patterns With Semantic Web Technology. Proceedings of CHI Workshop on Perspectives on HCI Patterns: Concepts and Tools

Herbsleb J., Zubrow, D., Goldenson, D., Hayes, W. and Paulk, M., 1997. Software Quality and Capability Maturity Model. Communications of the ACM. 40(6), 30-40

Hooper, K., 1986. Architectural Design: An Analogy. In D.A. Norman and S.W. Draper (Eds.), User Centered System Design: New perspectives on Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Erlbaum Associates

I

IEEE/ACM, 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering: IEEE/ACM. The Joint Task Force on Computing Curricula, IEEE Computer Society Association for Computing Machinery. Available at <http://www.acm.org/education/curricula-recommendations>

Infragistics, n.d. Quince – UX Design Patterns. Available at <http://quince.infragistics.com/ux-design-patterns.aspx> accessed July 2009

J

Jonassen, D. and Carr, C., 2000. Mindtools: Affording Multiple Knowledge Representations for Learning. Computers as cognitive tools: No more walls. Lawrence Erlbaum Associates, 2

Jonassen, D., 1999. Designing Constructivist Learning Environments. Instructional design theories and models: A new paradigm of instruction theory. Lawrence Erlbaum Associates, 11, 215-139

Jonassen, D., Myers, J., and McKillop, A., 1996. From Constructivism to Constructionism: Learning with hypermedia/multimedia rather than from it. Constructivist learning environments: Case studies in instructional design. Englewood Cliffs: Educational Technology Publications, 93-106

K

Kamthon, P., 2006. A critique of Pattern Language Markup Language (PLML). Interfaces: Journal for the British HCI Group, 68(Autumn), 14-15

Kaplan, B. and Duchon, D., 1988. Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study. MIS Quarterly, 12(4), 571-586

Kauloi, M. and Karlsson, I., 1998. Triangulation strategies in user requirements investigations: a case study on the development of an IT-mediated service. Behaviour and Information Technology, 17(2), 103-112

Kemp, R., Todd, E., Krsinich, R. and Elgar, C., 2007. Bridging the Gap between Observing and Doing: Facilitating the learning of OO Programming. Journal of Technology, Instruction, Cognition and Learning, 5(1), 13-39

Kemp, R., Kemp, E. and Todd E., 2009. Self-regulated Fading in On-line Learning. The 14 International Conference on Artificial Intelligence in Education (AIED2009), Brighton, UK

Kirakowski, J., 1994. The Use of Questionnaire Methods for Usability Assessment. SUMI: Software Usability Measurement Inventory, Human Factors Research Group, Ireland. Available at <http://www.ucc.ie/hfrg/questionnaires/sumi/sumipapp.html> accessed March 2010

Kitchenham B. and Pickard, L., 1998. Evaluating Software engineering methods and Tools. Part 9: Quantitative Case Study Methodology. ACM SIGSOFT Software Engineering Notes 23(1), 24-26

Kitchenham, B. and Pfleeger, S., 2002a. Principles of Survey Research Part 3: Constructing a Survey Instrument. Software Engineering Notes, 27(2), 20-24

Kitchenham, B., Pfleeger, S., Pickard, L., Jones. P., Hoaglin, D., El Emam K. and Rosenberg, J., 2002b. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, 28(8), 721-728

Kolfschoten, G., Lukosch, S., Verfraeck, A., Valentin, E. and de Vreede, G., 2010. Cognitive learning efficiency through the use of design patterns in teaching. Computers and Education 54, 652-660

Kotzé, P., Renaud, K. and van Biljon, J., 2008. Don't do this-Pitfalls in using anti-patterns in teaching human-computer interaction principles. Computers and Education 50(3), 979-1008

Kotzé, P., Renaud, K., Koukouletsos, K., Khazaei, B. and Dearden, A., 2006. Patterns, Anti-Patterns and Guidelines – Effective Aids to Teaching HCI Principles?. Proceedings of HCI Education: Inventivity, Teaching theory, design and innovation in HCI, The Lakeside Hotel, Ballina/Killaloe, Ireland, 10.11-10.17

Koukouletsos, K., 2008. Empirical Assessment of Patterns and Guidelines for Web Design. Unpublished PhD, Sheffield Hallam University, Sheffield

Koukouletsos, K., Khazaei, B., Dearden, A. and Ozcan, M., 2007. Teaching Usability Principles with Patterns and Guidelines. HCIEducators, Aveiro, Portugal, 67-74

Koukouletsos, K., Khazaei, B., Dearden, A. and Ozcan, M., 2009. Teaching Usability Principles with Patterns and Guidelines. Creativity and HCI: From Experience to Design in Education, Boston: Springer, 289, 159-174

Kruschitz, C., 2009. XPLML: a HCI pattern formalizing and unifying approach. Spotlight on Works in Progress: Proceedings of the Conference on Human Factors in Computing Systems. Boston, MA, USA: ACM. 4117-4122

L

Laakso, S.A., 2003. Introduction to User Interface Design Patterns. Available at <http://www.cs.helsinki.fi/u/salaakso/patterns/> accessed July 2007

Laakso, S.A., n.d. User Interface Design Patterns. Available at <http://www.cs.helsinki.fi/u/salaakso/patterns/> accessed July 2007

Laakso, S. and Laakso, K., 2004. Ensuring quality of the user interface with the GUIDe project model. Available at <http://www.cs.helsinki.fi/u/salaakso/> accessed December 2004

Laakso, S., Laakso, K. and Saura, A., 2000. Pattern Submission to CHI 2000 Workshop. Proceedings of The Future is Here-Pattern Languages for Interaction Design Workshop: Building Momentum, Hague, Netherlands

Leacock, M., Malone, E. and Wheeler, C., 2005. Implementing a Pattern Library in the Real World: A Yahoo! Case Study. Proceedings of Information Architecture Summit — Crossing Boundaries, Fairmont The Queen Elizabeth, Montréal, Québec, Canada. American Society for Information Science and Technology

Leacock, M., Malone, E. and Wheeler, C., n.d. Yahoo! Design Pattern Library.. Available at <http://developer.yahoo.com/ypatterns/index.php> accessed January 2007

Lewis, T., Rosson, M. and Pérez-Quiñones, M., 2004. What Do The Experts Say? Teaching Introductory Design from an Expert's Perspective. SIGCSE'04, 296-300

Lin, J. and Landay J., 2003. Damask: A Tool for Early-Stage Design and Prototyping of Cross-Device User Interfaces. CHI 2003 Workshop: Perspectives on HCI Patterns: Concepts and Tools. Fort Lauderdale, Florida USA, ACM

Lin, J., and Landay, J., 2008. Employing Patterns and Layers for Early-Stage Design and Prototyping of Cross-Device User Interfaces. Proceedings of CHI 2008: Human Factors in Computing Systems. Florence, Italy, 1313-1322

Lin, J., Thomsen, M. and Landay, J., 2002. A visual language for sketching large and complex interactive designs. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves. Minneapolis, Minnesota, USA. ACM Press New York, NY, USA, 307-314

Little, A., 2009. Introducing Quince – A UX Design Pattern Explorer. Available at <http://community.infragistics.com/blogs/ux/archive/2009/02/02/introducing-quince-a-ux-design-pattern-explorer.aspx> accessed April 2010

Lockamy III, A. and McCormack, K., 2004. The development of a supply chain management process maturity model using the concepts of business process orientation. International Journal of Supply Chain Management: 9(4),·272-278

Lukosch, S. and Schümmer, T., 2006. Foreword. Int. J. Computer Applications in Technology. 25(2/3), 85

M

Mahemoff, M., 1999. Incorporating Usability into the Software Process. Proceedings of Workshop-Usability Pattern Language: creating a community, Edinburgh, Scotland. British Computer Society

Mahemoff, M. and Johnston, L., 1998. Principles for a Usability-Oriented Pattern Language. In Proceedings of the Australian Computer Human Interaction Conference. Adelaide: IEEE Computer Society, Los Alamitos, 132—139

Mahemoff, M. and Johnston, L., 2001. Usability Pattern Languages: the "Language" Aspect. In Proceedings of Human-Computer Interaction, Tokyo, Japan. IOS Press (For IFIP), 350-358

Marcus, A., 2004. Fast forward: Patterns within Patterns. Interactions, 11(2), 28 – 34

Märtin C. and Roski A., 2007. Structurally Supported Design of HCI Pattern Languages. Human-Computer Interaction, Part I, HCII 2007, LNCS 4550, 1159–1167

McInerney, P., 2002. Patterns Workshop Report. Patterns Workshop, CHI2002. Philadelphia.

Merrill, M., 2002. First Principles of Instruction. ETR&D, 50(3), 43-59

Mingers, J., 2001. Combining IS Research Methods: Towards a Pluralist Methodology. Information Systems Research, 12(3), 240-259

Molina, P. and Hernandez, J., 2003. Just UI: Using patterns as concepts for IU specification and code generation. New Horizons: Perspectives on HCI Patterns: Concepts and Tools workshop, Fort Lauderdale, Florida, USA. ACM

Molina, P., Torres-Boigues, I., and Pastor, O., 2003. User Interface Patterns for Object-Oriented Navigation. UPGRADE: The European Online Magazine for the IT Professional, IV(1), 31-37

Muller, O., 2005. Pattern oriented instruction and the enhancement of analogical reasoning, Proceedings of the 2005 International Computing Education Research. ACM, Seattle, WA, USA, 57 – 67

Mullet, K., 2002. *Structuring Pattern Languages to Facilitate Design*, Changing the World, Changing Ourselves-Patterns in Practice workshop. Minneapolis, Minnesota, USA.: ACM

N

Noble, J., 1998. Classifying Relationships between Object-Oriented Design Patterns. In Proceedings of Australian Software Engineering Conference, Adelaide, SA. IEEE, 98-107

Norman, G., 2009. Editorial: When will learning style go out of style? Advances in Health Science Education, 14, 1–4

P

Patton, M. Q. 1990. Qualitative evaluation and research methods (2nd ed.). Thousand Oaks, CA, US: Sage Publications, Inc., 1990

Pemberton, L., 2000. The Promise of Pattern Languages for Interaction Design. Human Factors Symposium, Loughborough, UK. Available at <http://www.it.bton.ac.uk/staff /lp22/HF2000.html>, accessed Oct 2003

Pfleeger, S., 1994. Experimental Design and Analysis in Software Engineering Part 1: The Language of Case Studies and Formal Experiments. ACM SIGSOFT Software Engineering Notes, 19(4), 16-20

Pfleeger, S., 1995a. Experimental Design and Analysis in Software Engineering: Part 2: How to Set Up an Experiment. Software Engineering Notes, ACM SIGSOFT, 20(1), 22-26

Pfleeger, S., 1995b. Experimental Design and Analysis in Software Engineering Part 4: Choosing an Experimental Design. Software Engineering Note, ACM SIGSOFT, 20(3), 13-15

Pfleeger, S., 2005. Soup or Art? The Role of Evidential Force in Empirical Software Engineering. IEEE SOFTWARE, 22(1), 66-73

Pfleeger S. and Kitchenham B., 2002. Principles of Survey Research Part 1: Constructing a Survey Instrument. Software Engineering Notes, 26(6), 16-18

Phillips, C. and Joe, R., 2005. Supporting Content Design of Interaction Spaces. In Proceedings of the 6th Annual Conference of the NZ ACM SIGCHI: Making CHI Natural, Auckland, New Zealand. ACM Press. 39-44

Pribeanu, C. and Vanderdonckt, J., 2003. A Pattern-Based Approach to User Interface Development. New Horizons: Perspectives on HCI Patterns: Concepts and Tools workshop, Fort Lauderdale, Florida, USA. ACM

Prince, P., 2004. Does Active Learning Work? A Review of the Research, Journal of Engineering Education. 93(3), 223-231

Preece, J., Rogers, Y. and Sharp, H., 2002. Interaction Design: Beyond Human-Computer Interaction. New York, NY: John Wiley and Sons.

R

Ragan C., Nagel J. and White P., (Eds.) 2004. Scientific Foundations of Qualitative Research. Workshop report. National Science Foundation Report. Virginia, July 11-12, 1-148

Rettig, M., 1994. Prototyping for Tiny Fingers. Communications of the ACM. 37(4), 21-27

Rosenshine, B., 1970. The Stability of Teacher Effects Upon Student Achievement. Review of Educational Research. 40, 647-662

S

Salingaros, N., 2000. The Structure of Pattern Languages. Architectural Research Quarterly, 4, 149-161

Saponas, T., Prabaker, M., Abowd, G., and Landay, J., 2006. The impact of pre-patterns on the design of digital home applications. The conference on Designing interactive systems: processes, practices, methods, and techniques. University Park, Pennsylvania, USA: ACM New York, NY, USA, 189-198

Saunders, W., 2002. Review of "A Pattern Language". Harvard Design Magazine. Winter/Spring(16), 1-7

Savery, J. and Duffy, T., 2001. Problem based Learning: An instructional model and its constructivist framework. Center for Research on Learning and Technology. Indiana University, CRLT Technical report 16-01

Schmidt, D., Johnson, R. and Fayad, M., 1996. Software Patterns. Communications of the ACM, Special Issue on Patterns and Pattern Languages, 39(10), 37-39

Schümmer, T., 2003a. Evolving a Groupware Pattern Language. Proceedings of 8th European Conference of Computer-supported Cooperative Work, Workshop 6: From Good Practices to Patterns. Helsinki

Schümmer, T., 2003b. Seeking for structure in a Groupware Pattern Language. New Horizons workshop: Perspectives on HCI Patterns: Concepts and Tools, Fort Lauderdale, Florida, USA. 5, 4

Schümmer, T. and Lukosch, S., 2007. Patterns for Computer-Mediated Interaction. John Wiley and Sons

Schümmer, T. and Slagter R., 2004. The Oregon Software Development Process. In Proceedings of Fifth International Conference on eXtreme Programming and Agile Processes in Software Engineering, Garmisch-Partenkirchen, Germany

Schümmer, T., Lukosch, S. and Slagter, R., 2006. Using patterns to empower end-users- The Oregon Software Development Process for groupware. International Journal of Cooperative Information Systems, 15(2), 259-288

Seffah, A., 2003. Learning the ropes: human-centered design skills and patterns for software engineers' education. Interactions, 10(5), 36-45

Seffah, A. and Forbrig, P., 2002. Multiple User Interfaces: Towards a Task-Driven and Patterns-Oriented Design Model. In Forbrig, P., Limbourg, Q., Urban, B., Vanderdonckt, J. (eds.) DSV-IS 2002. Springer, Heidelberg. LNCS, 2545, 118–132

Seffah, A. and Gaffer, A., 2007. Model-based user interface engineering with design patterns. Journal of Systems and Software, 80(8), 1408-1422

Segerstahl, K. and Jokela, T., 2006. Usability of interaction patterns. Proceedings of Conference on Human Factors in Computing Systems: Interact, Inform, Inspire, Montréal, Québec, Canada. ACM Press, New York, NY, USA, 1301 – 1306

Sharp, H., Manns, M. and Eckstein, J., 2003. Evolving Pedagogical Patterns: The Work of the Pedagogical Patterns Project. Computer Science Education, 13(4), 315-330

Shavelson, R., Phillips D., Towne, L. and Feuer, M., 2003. On the Science of Education Design Studies. Educational Researcher, 32(1), 25-28

Sherwood, A. and Phillips, J., n.d. Rank of Coal. Te Ara Encyclopedia of New Zealand. Available at <http://www.teara.govt.nz/en/coal-and-coal-mining/1/2> updated 2009, accessed March 2010

Shinichi, H., Yasuyuki, T., Nobukazu, Y., Kenji, T. and Hironori, W., 2007. Top SE: Educating Superarchitects Who Can Apply Software Engineering Tools to Practical Development in Japan. Proceedings of the 29th international conference on Software Engineering: IEEE Computer Society

Shneiderman, B., 1998. Designing the User Interface: strategies for effective human-computer-interaction. 3rd ed. Addison Wesley Longman, Inc

Software Engineering Institute, n.d. Capability Maturity Model Integration (CMMI) Carnegie Mellon, Available at < http://www.sei.cmu.edu/cmmi/index.cfm> accessed March 2010

Sinnig, D., Javahery, H., Forbrig, P. and Seffah, A., 2003. The Complicity of Model-Based Approaches and Patterns for UI Engineering. Proceedings of Business Informatics Research. Berlin, 120-131

Snow, K., Marks, M., Hong, D. and Dennis, T., 2006. Berkley Web Patterns Project. Available at <http://www.ui-designpatterns.org/index.html> accessed July 2007

Suggate R., 2000. Application of Rank(Sr), a maturity index based on chemical analyses of coals. Marine and Petroleum Geology. 19, 929–950

T

Tidwell, J., 1999. Common Ground: A Pattern Language for Human-Computer Interface Design: The Case for HCI Design Patterns. Available at <http://www.mit.edu/~jtidwell/interaction_patterns.html> accessed August 2002

Tidwell, J., 1998. Interaction Design Patterns. PloP'98 Conference on Pattern Languages of Programming. Illinois, USA

Tidwell, J., 2006. Designing interfaces. Sebastopol, CA: O'Reilly

Tidwell, J., n.d. UI Patterns and Techniques. Available at <http://designinginterfaces.com/Visual_Framework> accessed December 2006

Todd, E., 2004. Using Patterns to Guide User Interface Development. The 6th Asian Pacific Conference on Computer Human Interaction. Rotorua, N.Z. 682-686

Todd, E., Kemp, E., and Philips, C., 2003. Validating user Interface Pattern Languages, Computer Human Interface New Zealand. University of Otago, Dunedin, NZ:ACM SIGCHI, 125-126

Todd, E., Kemp, E., and Philips, C., 2004. What makes a good User Interface pattern language?, The 5th Australasian User Interface Conference. Dunedin: Australian Computer Society, 28, 91-100

Todd, E., Kemp, E., and Philips, C., 2009. Introducing Students to UI patterns, Proceedings of the 10th Conference on Human-Computer Interaction, CHINZ'09. Auckland, New Zealand: ACM, 37-40

Trochim, W., 2006. The Research Methods Knowledge Base, 2nd Edition. Available at <http://www.socialresearchmethods.net/kb/> accessed February 2009

V

van Biljon, J., Kotzé, P., Renaud, K., McGee, M. and Seffah, A., 2004. The Use of Anti-Patterns in Human Computer Interaction: Wise or Ill-Advised?. Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries. Protea Hotel, Stellenbosch. 176-185

van der Stuyf, R., 2002. Scaffolding as a Teaching Strategy. Adolescent Learning and Development, Section 0500A-Fall. Available at <http://condor.admin.ccny.cuny.edu/~group4/> accessed January 2010 van Duyne, D., Landay, J. and Hong, J., 2003. The Design of Sites-Patterns principles and processes for crafting a customer-centred web experience. Boston, MA: Addison-Wesley

van Welie, M., 2001. Task-based User Interface Design. Unpublished PhD, Vrije Universiteit

van Welie, M. and Klaassen, B., 2004. Evaluating Museum Websites using Design Patterns International workshop on Tools for Working with Working Guidelines. Biarritz, France. Report No. IR-IMSE-001

van Welie, M., and van der Veer, G., 2003. Pattern Languages in Interaction Design: Structure and Organization. Proceedings of Interact: Bringing the Bits Together. Zürich, Switzerland: IOS Press. 527-534

van Welie, M. and Traetteberg, H., 2000. Interaction Patterns and User Interfaces. In Proceedings of Pattern Languages of Programming. Allerton Park, Monticello, Illinois, USA

van Welie, M., n.d. Patterns in Interaction Design. Available at <http://www.welie.com/index.php> accessed January 2007

W

Walldius, A., 2003. Patterns can bring in knowledgeable users into the software development cycle. Proceedings of 2nd Workshop on Software and Usability Cross-Pollination: The Role of Usability Patterns, Zürich, Switzerland

Wania, C. and Atwood, D., 2009. Pattern languages in the wild: exploring pattern languages in the laboratory and in the real world. Proceedings of the 4th International Conference on Design Science Research in Information Systems, May 07 – 08, Philadelphia, Pennsylvania. ACM

Wania, C.E., 2008. Examining the Impact of an Information Retrieval Pattern Language on the Design of Information Retrieval Interfaces, Unpublished PhD, Drexel University, USA

Weiss, S., 2005. Teaching design patterns by stealth. Proceedings of SIG Computer Science Education. St. Louis, Missouri, USA.ACM Press New York, NY, USA. 492-494

Wentzlaff, I. and Specker, M., 2006. Pattern-based development of user-friendly web applications. Proceedings of Workshop of the 6th International Conference on Web Engineering. Palo Alto, California

Wesson, J., and Cowley, L., 2003. Designing with Patterns: Possibilities and Pitfalls, Bringing the Bits Together: Proceedings of the 2nd Workshop on Software and Usability Cross-Pollination-The Role of Usability Patterns. Zürich, Switzerland.

Wilkins, B., 2003. MELD: A Pattern Supported Methodology for Visualisation Design. Unpublished Doctor of Philosophy, The University of Birmingham

Wilson, C. E., 2006. "Triangulation: The Explicit Use of Multiple Methods, Measures and Approaches for Determining Core Issues in Product Development". Interactions, 63(Nov-Dec), 46-47

Wood, D., 2001. Scaffolding, contingent tutoring and computer-supported learning. International Journal of Artificial Intelligence in Education. 12, 280-292

Wood, D. and Wood, H., 1996. Vygotsky, Tutoring and Learning. Oxford Review of Education. 22(1), 5-16

Wood, D., Bruner, J., and Ross, G., 1976. The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry. 17(2), 89-100

Y

Yahoo, n.d. Yahoo! Design Pattern Library. Available at: <http://developer.yahoo.com/ypatterns/index.php> accessed January 2007

Yin, R., 1994. Case Study Research: Design And Methods (2nd ed.). Thousand Oaks: Sage Publications

Yourdon, E., 1989. Modern Structured Analysis. Prentice-Hall, Englewood Cliffs

Publications

Directly related to this research

Todd, E., 2004. Using Patterns to Guide User Interface Development. The 6th Asian Pacific Conference on Computer Human Interaction. Rotorua, N.Z. 682-686

Todd, E., Kemp, E., and Philips, C., 2003. Validating user Interface Pattern Languages, Computer Human Interface New Zealand. University of Otago, Dunedin, NZ:ACM SIGCHI, 125-126

Todd, E., Kemp, E., and Philips, C., 2004. What makes a good User Interface pattern language? The 5th Australasian User Interface Conference. Dunedin: Australian Computer Society, 28, 91-100

Todd, E., Kemp, E., and Philips, C., 2009. Introducing Students to UI patterns, Proceedings of the 10th International Conference NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction (SIGCHI-NZ), CHINZ 2009. Auckland, New Zealand: ACM, 37-40

Todd, E., Kemp, E., and Philips, C., 2010. Guiding the development of UI pattern models in an educational context. Proceedings of the 11th International Conference NZ Chapter of the ACM Special Interest Group on Human-Computer Interaction (SIGCHI-NZ), CHINZ 2010. Auckland, New Zealand: ACM, 57-64

Associated research

Deng, J., Kemp, E. and Todd, E., 2005. Managing UI Pattern Collections. Proceedings of 6th Annual Conference of the NZ ACM SIGCHI: Making CHI Natural: Auckland, New Zealand

Deng, J., Kemp, E. and Todd, E., 2006. Focussing on a standard pattern form: The development and evaluation of MUIP. Proceedings of 7th Annual Conference of the NZ ACM SIGCHI: Design Centered HCI, University of Canterbury, Christchurch, New Zealand. ACM press

Kemp, R., Todd, E., & Lu, J. Y. 2003. Designing a flexible learning environment: Learning from books. In U. Hoppe, F. Verdejo & J. Kay (Eds.), Artificial intelligence in education: Shaping the future of learning through intelligent technologies (Frontiers in Artificial Intelligence and Applications, 434-436). Amsterdam, Netherlands: IOS Press.

Kemp, R., Todd, E., Krsinich, R. and Elgar, C., 2007. Bridging the Gap between Observing and Doing: Facilitating the learning of OO Programming. Journal of Technology, Instruction, Cognition and Learning, 5(1), 13-39

Kemp, R., Kemp, E. and Todd E., 2009. Self-regulated Fading in On-line Learning. The 14 International Conference on Artificial Intelligence in Education (AIED2009), Brighton, UK

Learning about User Interface Design through the use of User Interface Pattern Languages

Volume 2 - Appendices

Elisabeth-Ann Gynn Todd

2010

Contents

Appendix A1: Pattern Language Maturity Model

Level	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**
	Validation Map	**Hierarchy**	**Abstraction**	**Match**	**Complexity**	**Richness**	**Views**
Immature (0)	None Minimal <50%	None Minimal	None Minimal	None Minimal <20%	None / Minimal <5%	None	None
Partially mature (1)	Possibly / Developing <70%	Possibly / Developing	Possibly/ Developing	Developing <=50%	Possibly / Developing <25%	Possibly / Developing <10%	Possibly / Developing 1
Moderately mature (2)	Developed <90%	Developed 2 levels	Developed	Developed >50%	Developed <35%	Developed <20%	Developed 2
Fully mature (3)	>=90% Mapped	>=3 levels Hierarchical	Modelled	>=80% Matched	>=35% Complex	>=20% Rich	>2 Views

Table A1.1- Maturity level goals for each UI pattern language structural attribute

The set of rules that form the tests for determining a pattern language's maturity status are described below. The scales for the tests are used to establish the maturity of pattern languages and can be used to compare pattern languages and to calculate a pattern language maturity index.

A1.1 Maturity Model Tests

When assessing internal validity of a UI pattern language results from primary structure tests are evaluated as a pass or fail but are assigned a value when used to assess maturity. Secondary structures relate to how complicated the UI pattern language's structure is becoming.

Primary Structure Tests - Internal Validity

1. Do the references and contexts linking the patterns form a validation pattern map including at least ninety percent of the patterns?
   (None, Minimal, Possibly, Developing, Developed, Mapped)

2. Can the map be ordered into at least one hierarchy with at least three levels below root level?
   (None, Minimal, Possibly, Developing, Developed, Hierarchical)

3. Can the layers making up the hierarchy be used to describe a UI at different levels of granularity?
   This can be demonstrated by creating a UI-pattern model for an example UI.
   (None, Minimal, Possibly, Developing, Developed, Modelled)

4. Is the difference between the context map and the idealised map less than twenty percent and similarly between the reference map and the idealised map?
   (None, Minimal, Developing, Developed, Matched)

Secondary Structure Tests

5. How 'complex' are the links within and across levels of the hierarchy?
   (*None, Minimal, Possibly, Developing, Developed, Complex*)
6. How 'rich' are the relationships linking the patterns?
   (*None, Possibly, Developing, Developed, Rich*)
7. How many alternative views can be identified when using this pattern language?
   (*None, Possibly, Developing, Developed, Views*)

A1.2 Scales for Assessing Maturity Criteria

The scales for each of these measures are qualified by how connected the patterns in the pattern language being assessed are. Two forms of connectivity are considered. First, for the primary structures the connectivity of the validated pattern map is important. The validation pattern map is restricted to just those links representing the basic relationship types. Secondly, some measure of the connectivity of the pattern map which includes links of any relationship types is important. Connectivity is the percentage of the patterns in the pattern language that are linked to one or more of the other patterns in the language.

Measurement Scales for Primary Structures

Only basic relationships are included. They are: *contained* and the inverse is-*contained-by* and the generalisation-specialisation relationship *is-a*. For the Hierarchy, Abstraction and Match scales the measure of ‘Developing’ requires the connectivity of the validated pattern map to be at least seventy percent. The measure of ‘Developed’ requires the connectivity of the validated pattern map to be at least ninety percent.

1. **Validation Map Scale**

None [<10%] – there is less than ten percent connectivity between patterns.

Minimal [<50%] – connectivity of the validation pattern map is less than fifty percent

Possibly [>80% undefined link types] – connectivity of the pattern map is at least eighty percent but patterns don’t indicate relationship types. Alternatively no detailed analysis has been made of the patterns in the language so relationship types remain unclear.

Developing [<70%] – connectivity of the validation pattern map is less than seventy percent.

Developed [<90%] – connectivity of the validation pattern map is less than ninety percent.

Mapped [>=90%] – connectivity of the validation pattern map is at least ninety percent.

2. Hierarchy Scale

None [<50% connectivity] – Not sufficiently connected to determine valid hierarchies

Minimal [<70% connectivity + 1 level parent] – connectivity of the validation pattern map is less than seventy percent. One or more partial hierarchies are clearly identifiable with at least one layer below an identifiable parent.

Possibly [>80% undefined link types + partial] – connectivity of the pattern map is greater than eighty percent. Some apparent hierarchies of patterns can be identified but relationship types are unclear.

Developing [<90% connectivity + 2 levels parent] – connectivity of the validation pattern map is less than ninety percent. One or more partial hierarchies are clearly identifiable with at least two layers below an identifiable parent.

Developed [>=90% connectivity + 2 levels root] – connectivity of the validation pattern map is at least ninety percent. It can be ordered into one or more hierarchies where at least one hierarchy has at least two levels below a clear root.

Hierarchical [>=90% connectivity + >= 3 levels root] – connectivity of the validation pattern map is at least ninety percent. It can be ordered into one or more hierarchies where at least one hierarchy has at least three levels below a clear root.

3. Abstraction Scale

None [<50% connectivity] – Not connected enough or no usable levels of granularity can be identified.

Minimal [<70%] – connectivity of the validation pattern map is less than seventy percent. Some groups of patterns in identifiable in partial hierarchies are at the same level of granularity.

Possibly [<80% undefined link types] – connectivity of the pattern map is greater than eighty percent. Relationships are unclear but some groups of patterns appear to be at the same level of granularity and can be used as a generalised description of a UI.

Developing [>90%] – connectivity of the validation pattern map is at least seventy percent. Some groups of patterns in identifiable hierarchies are at the same level of granular**it**y and can be used as a generalised description of a UI.

Developed [>=90%] – connectivity of the validation pattern map is at least ninety percent. The layers making up the hierarchy can easily be used to describe a UI at different levels of granularity.

Modelled [>=90% plus demo] – connectivity of the validation pattern map is at least ninety percent. Levels of granularity demonstrated with UI examples, e.g. UI-pattern models.

4. Match Scale

None [<70 connectivity] – Not connected enough. There are no matches or context and reference maps cannot be created

Minimal [<90% connectivity + <20%] – the number of context and reference links that match is less than twenty percent.

Developing [<90% connectivity + <50% match] – connectivity of the validation pattern map is less that ninety percent, and/or the number of context and reference links that match is less than fifty percent.

Developed [>=90% connectivity + <80% match] – connectivity of the validation pattern map is at least ninety percent. The number of context and reference links that match is less than eighty percent.

Matched [>=90% connectivity + >=80% match] – connectivity of the validation pattern map is at least ninety percent and the number of context and reference links that match is at least eighty percent.

Measurement Scales for Secondary Structures

For the Complexity, Richness and View scales the measure of ‘Developing’ and above require the connectivity of the pattern map to be at least ninety percent.

Complexity Scale

This scale relates to firstly, the number of inter-level links, that is the number of links between patterns on the same level in the underlying hierarchical structure and secondly, the number of intra-level links which are links that cross over more than one level of the underlying hierarchy.

None – no complete hierarchies and/or so many undifferentiated links the user of the pattern language becomes confused.

Minimal [>=80% connectivity + <5%] – connectivity of the pattern map is greater than eighty percent and one or more hierarchies can be identified but less than five percent of the links are intra-level links and/or inter-level links identified.

Possibly [>=80% connectivity +>50%] – connectivity of the pattern map is greater than eighty percent. But more than fifty percent of the links are intra-level and inter-level so that it is excessively complex.

Developing [>=90% connectivity + >=1 hierarchy + <25%] – at least ninety percent of the patterns are linked and one or more hierarchies can be identified. But less that twenty-five percent of the links within at least one level are intra-level links and some inter-level links identified.

Developed [>=90% connectivity + >=1 hierarchy + <35%] – at least ninety percent of the patterns are linked and one or more hierarchies can be identified. But less than thirty-five percent of the links are intra-level links and inter-level links.

Complex [>=90% connectivity + >=1 hierarchy + >=35%] – at least ninety percent of the patterns are linked and one or more hierarchies can be identified. More than thirty-five percent of the links are intra-level links and occur in more than half of the levels below the root.

Richness Scale

None – non-basic types of relationship cannot be identified.

Possibly [>=80% connectivity + some non-basic links] – connectivity of the pattern map is greater than eighty percent, and some non-basic relationship types have been identified

Developing [>=90% connectivity + <10% non-basic links] – some, but less than ten percent of the relationships are of non-basic types.

Developed [>=90% connectivity + >=10% non-basic links] – at least ten percent of all relationships are of non-basic types

Rich [>=90% connectivity + >20% non-basic links] – over twenty percent of the relationships are of non-basic types.

View Scale

Scale is related to the primary structure hierarchy characteristic.

None – no useful partial hierarchies identified.

Possibly [>=80% connectivity + partial hierarchies] – connectivity of the pattern map is greater than eighty percent. More than one partial hierarchy with a specific focus can be identified

Developing [>=90% connectivity +1 hierarchy] – the pattern map can be organised into one hierarchical structure.

Developed [>=90% connectivity +2 hierarchy] – the pattern map can be organised into two overlapping hierarchical structures.

Views [>=90% connectivity +>2 hierarchy] – the pattern map can be organised into more than two overlapping hierarchical structures.

A1.3 Creating a maturity Index

The overall maturity rating for a pattern language is calculated by summing the scores for the four primary structures, and adding to that the sum for the three secondary structures multiplied by four over three so that the scores for both sets of structures have equal weighting.

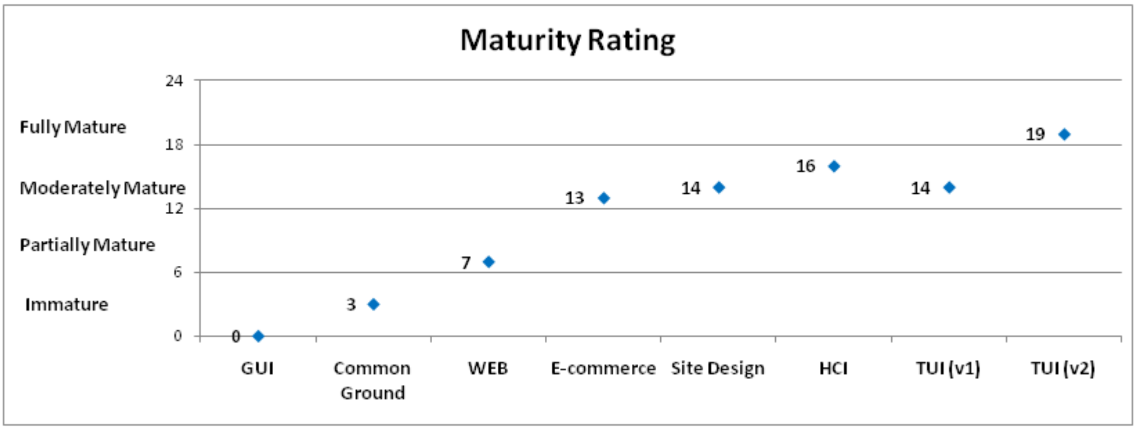
Pattern Language Rating = ROUND(SUM(Map + Hierarchy + Abstraction + Match)
+(SUM(Complexity + Richness + Views) * 4/3),0)

A1.4 Examples

Table of Attribute Values for Eight UI pattern languages

	Map	Hierarchy	Abstraction	Match	Complexity	Richness	Views	Rating
GUI	0	0	0	0	0	0	0	0.0
Common Ground	1	1	0	0	0	0	1	3.3
WEB	1	1	1	0	1	1	1	7.0
E-commerce	3	3	2	0	1	1	2	13.3
Site Design	3	3	1	0	1	1	3	13.7
HCI	3	3	3	3	1	1	1	16.0
TUI (v1)	3	3	3	1	1	1	1	14.0
TUI (v2)	3	3	3	3	3	1	1	18.7

Graph of Attribute Values for Eight UI pattern languages



Appendix A2: The UI-pattern Model

A2.1 UI-pattern modelling

1. Create a partial pattern map by linking selected patterns together.
2. Remove all links that could not be classified as a basic type.
3. Remove directional arrows from the arcs.
4. Identify potential roots.
5. Identify relevant links by creating an overlay and removing unnecessary relationships.
6. Redraw the graph of relevant links identifying nodes by both pattern id and name.
7. Identify any missing links.
8. Correct existing links.
9. Add duplicated patterns where they are required to describe a distinct feature of the UI.
10. Add labels to relevant arcs related to UI components and interaction spaces

A2.2 Example: Outlook Today view

The partial pattern map in Figure 3.3 showed the twenty-two patterns selected to represent the Outlook Today view using links depicting all relationships identified between the participant patterns plus those that could be inferred from the text. The following procedure describes how this model is converted into the first version of the UI-pattern model.

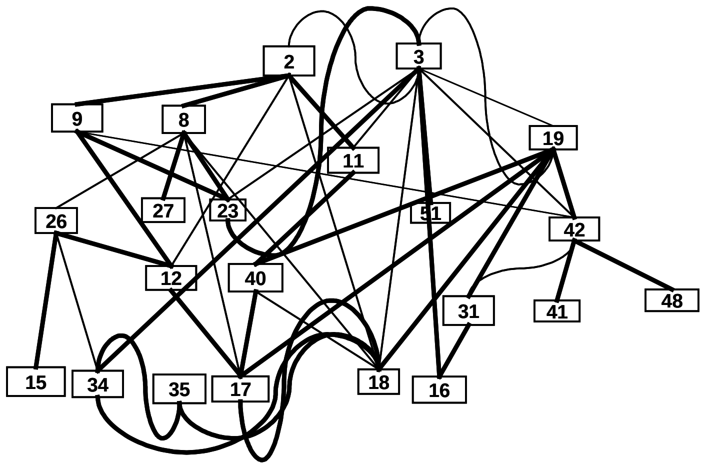


Figure A2.1 - Partial pattern map for Outlook's Today view using just basic relationships

The steps are:

1. Create a partial pattern map by linking selected patterns together.
2. Remove all links that could not be classified as a basic type.
3. Remove directional arrows from the arcs. This new diagram, shown in Figure A2.1, looks similar to a validated partial pattern map where all relationships matched.
4. Identify potential roots. Figure A2.1 shows that the only pattern with no context links is Pattern '34 Choice from a large set' which is clearly not a potential root. Study of pattern content indicates that Patterns '02 High-density info display' and '03 Status display' are general descriptions and are potential roots. Pattern 3 was rejected as the root because of the amount of information that needs to be displayed and the relatively complex structure of the information. Therefore Pattern 2 was selected.

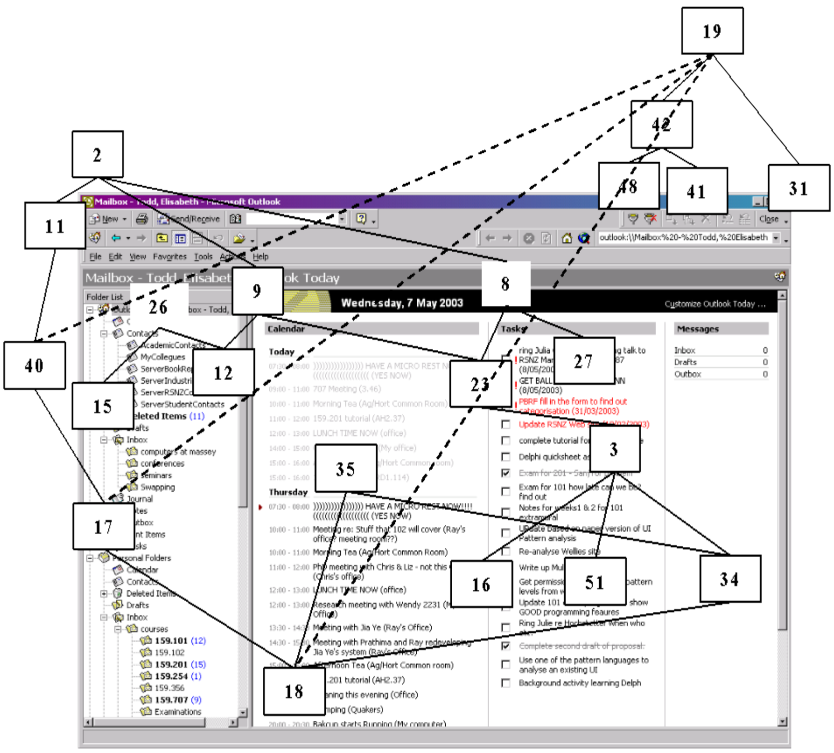


Figure A2.2 – Graph of patterns overlaying example UI

5. Identify relevant links by creating an overlay and removing unnecessary relationships. To create the overlay place the pattern identifying numbers (ids) near the parts of the example UI that they referred to as shown in Figure A2.2. Relationships between patterns that did not appear relevant to this example were removed. The resultant graph formed the initial backbone for the hierarchy building activity.

Many of the remaining relationships can be described as relating to a spatial component of the UI but temporal components were also identified. For example, the map shown in Figure A2.2, has a link directly between Pattern ‘40 Toolbox’ and Pattern ‘18 Short description’ but there is also a chain of links from Pattern 40 to Pattern ‘17 Pointer shows affordance’ and then from Pattern 17 to Pattern 18. When used in an actual situation the pointer will show affordance when coming into the space of a specific tool and a short description would appear in a user’s view. Therefore just the reference chain was retained: patterns 40 to 17 to 18.

6. Redraw the graph of relevant links identifying nodes by both pattern id and name. After re-examine each pattern’s context and reference sections, any links found to be required should be reinstated.

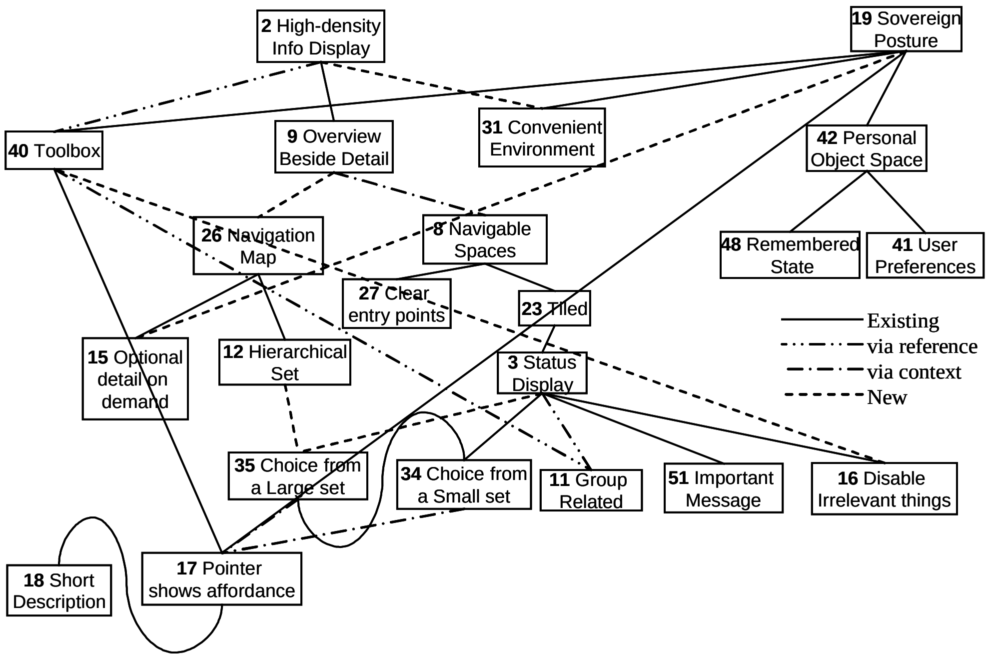


Figure A2.3 – Restructured partial pattern map for the Outlook example

7. Identify any missing links. Four kinds of relationship have been used to create the modified graph found in Figure A2.3. Solid lines represent the existing relationships. Based on observed behaviour and placement of widgets, six new relationships were added. These are shown as dotted lines. Figure 12 had two patterns with no context relationship linking them into the hierarchy: Pattern ‘34 Choice from a large set’ and Pattern ‘26 Navigation map’. New relationships were created to link them into the hierarchy. Pattern 35 has been linked to Pattern ‘10 Hierarchical Set’ so that items can be selected from such an organisation. Pattern 35 has also been linked to Pattern 3 ‘Status display’ to represent the set of entries in the

calendar, a long selectable list. The other new relationships were added so that descriptions at a specific level were more complete. For example, the link from Pattern ‘2 High-density information display’ to Pattern 31 ‘Convenient environment’ to represent the actions that can be carried out with the main window such as resize, close and minimise buttons.

8. Correct existing links. Corrections modifications required related to existing indirect links between the patterns either via reference links or via context links. For example, patterns 34 and 35 are both concerned with choices from sets and both had defined links to Pattern 18 ‘Short Description’ which links back-up to Pattern 17 ‘Pointer shows affordance’. For describing the behaviour observed in the example UI, linking patterns 35 and 34 directly to Pattern 17 was a more accurate description as the pointer would have to pass over a chosen item before the short description would appear.

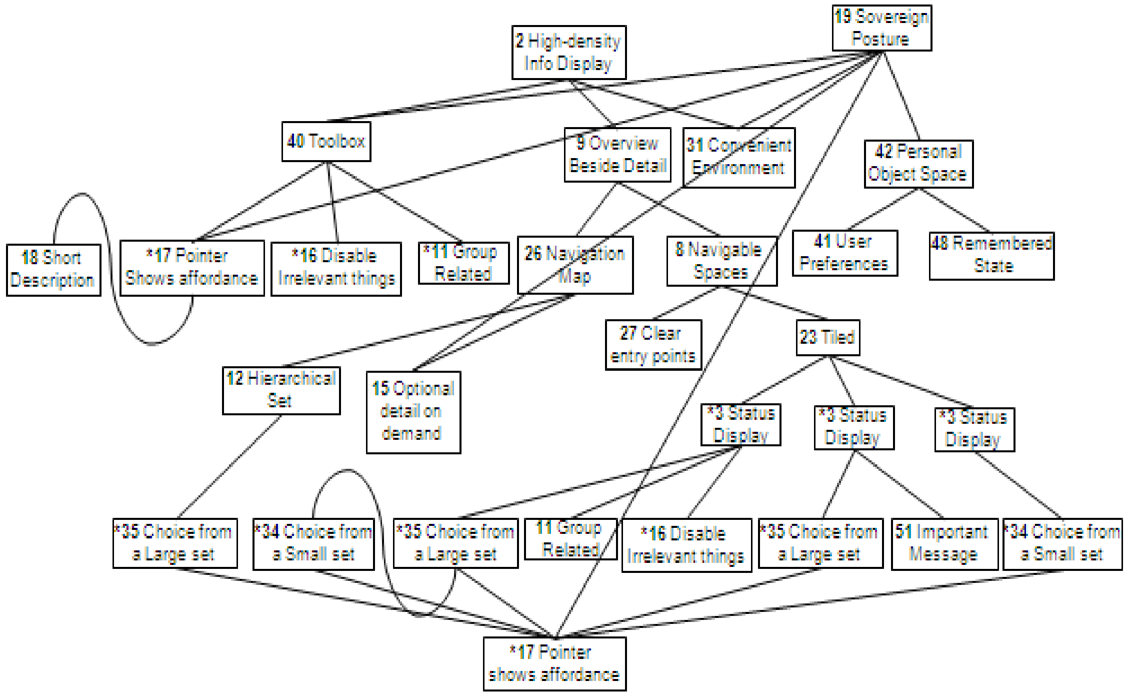


Figure A2.4 – Almost complete UI-pattern model for the Outlook example

9. Duplicated patterns where they are required to describe a distinct feature of the UI. Figure A2.4 shows duplicates are prefaced with an asterisk. At the lower levels, for example, Pattern 3 is repeated three times representing the spaces required for the Calendar information, Tasks and Messages respectively.

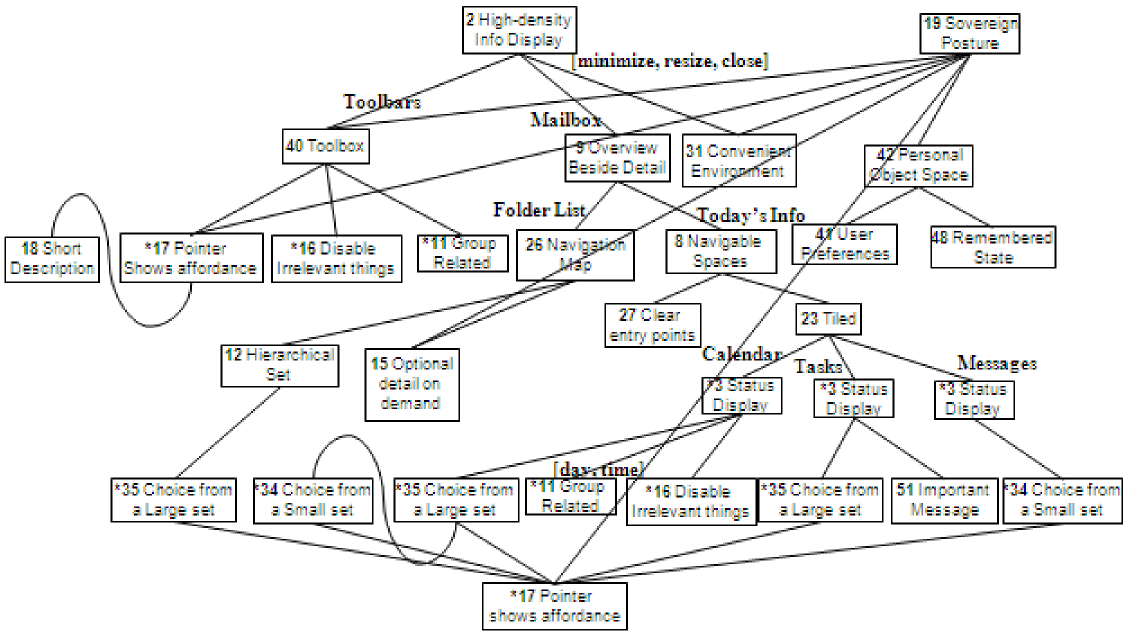


Figure A2.5 - Completed UI-pattern model with labelled arcs

10. Identify sections (sub-trees) that match UI interaction spaces. The descriptions generated by the levels and within sub-trees based on the UI-pattern model have a strong spatial component. For example, from level one the example screen from Outlook (Figure A2.2) can be described as having three main areas. These can be seen as labelled arcs in Figure A2.5. First Pattern 40 'Toolbox' representing the area were the toolbars are found. The second area represents the main structure of the information is represented by Pattern 9 'Overview beside detail' for the two main panels shown in the example. The third area represents convenient actions such as 'Minimise', 'Resize' and 'Close'.

Appendix A3: Studies One & Two – Pilot Studies

A3.1 The Pilot Studies for Study 1

Two pilot studies were run with participants who had completed the HCI course that research participants were enrolled in. The pilots were run to check both artefacts and procedures. The researcher acted as a pseudo-partner in each pilot but focussed on ensuring the research materials were understandable and procedures could be followed as expected. Checking timing was critical because the time available was limited.

The introduction was presented using paper copies of slides. Neither subject had any problems with the architectural example and one subject made comparisons with Software Engineering design patterns. Both participants identified grammar and comprehension problems with the instructions, questionnaires and pattern content. Modifications were made between the two pilot studies so those problems found in the second pilot were minimal. After completing the first exercise the first subject provided useful feedback that using the pattern names as part of the UI-pattern model made them more understandable than using just the pattern’s id number.

The researcher observed how the subject approached building the UI-pattern modelling exercise. No pattern list was provided for the first pilot and the subject suggested that this would be helpful. A copy was provided for the second pilot and this subject marked the pattern names on it as they used them, commenting that it was helpful to be able to see all the pattern names together.

The UI example used for both exercises in the first pilot proved to be too difficult to complete within the time period available. The examples were simplified for the second pilot and were completed within the time available.

A3.2 The Pilot Studies for Study 2

Two pilot studies were run to trial the changed experimental artefacts and procedures for Exercise One and the new set of artefacts and procedures for Exercise Two. One of the subjects had completed an HCI course. The second participant had the prerequisites for enrolling in the HCI course. This subject was selected because the lecturer had reservations about whether the students would be able to complete the TUIC modelling exercise without a more traditional introduction to the new material. The premise was that if a potential student who had not yet completed studying the HCI course could learn how to complete the TUIC model using the proposed method then the students enrolled in the course should not have any difficulty.

As with the pilot studies for Study Two the researcher acted as a pseudo partner in each of the pilots. Again timing was critical due to the two hour time limit available for the tutorial.

The first pilot was run with the subject with a background similar to the students. The protocol for this pilot included a pre-test for the TUIC modelling but it was clear that there was not enough time for this activity. The results indicating the subject did not know anything about TUIC modelling. It was eliminated. The TUIC post-test was shortened because the original test took too long to complete. This subject suggested that the pattern list include the lists of context and reference links.

The modified procedure was used with the second subject who only had the prerequisite background and they completed it within the time limit. The patterns list annotated with the context and reference link information was only used with the second exercise. No problems were encountered with the modified TUIC modelling questions. The second pilot subject found creating the UI-pattern model particularly demanding because they had difficulty interpreting some of the UI terms used in the narrative descriptions in the patterns. On the other hand they were very enthusiastic about the TUIC modelling exercise. They had no problems with the TUIC components test, scoring over ninety percent.

Observations of the subjects and comments they made while working with the patterns and the UI-pattern model as they created their TUIC models suggested changes that needed to be made to the TUIC model diagrams illustrating the solutions in some of the patterns. After working with the first subject alternative diagrams using both generic symbols and active materials were created for patterns 07, 10, 11 12 and 13 (Appendix 8). These changes were implemented and used with the second subject. During the TUIC modelling exercise it was clear that three other patterns should be modified to demonstrate different representations. These were hand-drawn on the patterns as modelling proceeded to explain the options available. The changes included providing examples of two ways of representing nested interaction spaces in patterns 03 and 06, and showing how one of the patterns used by another could be included in the solution for Pattern 08. A more explicit example for Pattern 14 showing a collection of actions was added as the subject had difficulty interpreting the repeat symbol.

Appendix A4: Studies One, Two & Three - Information Sheets

The following information sheets were given to potential participants in the different studies so that they could decide whether to participate. If they agreed to take part they then signed an appropriate Ethics Form as specified by Massey University's Human Ethics Committee.

The Information was designed to fit two sides of a sheet of A4 paper therefore they have been reproduced as images because the formatting for this document would not reproduce them correctly.

A4.1 Study One – Information Sheet

Using UI Patterns to Guide the Development of Conceptual User Interface Models

INFORMATION SHEET

All students are required to complete the exercises that make up this tutorial as part of the course requirements. Only responses from those students who agree to participate in the experiment will be used for this research. After filling in the consent forms, the forms will be immediately placed into a sealed envelope so the researcher and the research assistant who are taking this tutorial session will have no knowledge of who has or has not agreed to have their results included in the research.

Introduction

The overall purpose of this research is to investigate whether a user interface pattern language can be used to successfully guide novice UI developers in the creation of conceptual user interface models. The immediate investigation is primarily aimed at evaluating a method for introducing students to UI patterns with the aim of developing their understanding of pattern languages and the user interface knowledge contained within the patterns. As well the investigations will collect information about two presentation styles for presenting the information within UI patterns.

The goals of this experiment are:

- To collect information about a suitable pattern format for UI patterns to use with student UI designers.
- To evaluate whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model.

This research is being undertaken as part of doctoral studies by the researcher Elisabeth Todd. Supervision of this project is by Associate Professor Elizabeth Kemp and Associate Professor Chris Phillips of the Institute of Information Sciences and Technology.

Participant Recruitment

The activities necessary for this research will be conducted as part of the course for 159.353 - Human-Computer Interaction. Students enrolled in this paper are invited to have the analysis of their work included as part of the research.

The staff involved in teaching 159.353 will <u>not</u> know which students have agreed to allow their data to be included in the analysis of the experimental results. No student will be penalized for not agreeing to their data being included.

Project Procedures

The course lecturer will have no knowledge of who has agreed to have their results used in the research. Students' work will be marked by the course lecturer who is using assessment criteria independent to those used by the researcher. A copy of the student's work will then be made and passed to the researcher.

Only the researcher will collate the data. Once the data from individual students has been matched each set will be given an identifier then the cover sheets identifying individuals will be removed and destroyed. Data collected from individuals who have not consented to having their data included will be destroyed. The data collected will be held for the duration of the research and will be kept in a secure location and will be destroyed when the research is complete.

The data for this project will be collected in the following ways:

- The written results of the two pattern language modelling exercises will be analysed for the way the pattern models have been constructed;
- The observations of the researcher on the application of the method;
- Photographs of the work surfaces taken at timed intervals – All features that could be used to identify an individual will be removed;
- Questionnaires about the sets of patterns used in each exercise;
- A questionnaire about the method used and a comparison of the two types of patterns used.

The results from the analysis of the experimental data will be available from the CS office, AH2.76 on request. Students will have to present their ID card to prove that they were enrolled in this class and show they have a right to the information.

Participant involvement

As a participant in the experiment you will follow the same procedure for the tutorial as the other members of the class. The plan for the tutorial is as follows:

1. Introduction to explain the details of the study.
2. Complete ethics approval forms - All approval forms placed in a sealed envelope.
3. Students within each treatment group choose working partner from that group.
4. Complete Exercise One.
5. Fill in patterns questionnaire
6. Have a short break.
7. Complete Exercise Two.
8. Fill in patterns questionnaire
9. Answer exit questionnaire.

Participant's Rights

You are under no obligation to accept this invitation. If you decide to participate, you have the right to:

- Ask any questions about the research at any time during participation;
- Decline to answer any particular question related to the research;
- Decline to have your work surface photographed;
- Withdraw from the research before the end of the tutorial;
- Provide information on the understanding that all identifying information will be removed so that it becomes anonymous;
- Have access to a summary of the project findings when it is concluded.

Project Contacts

Researcher	**Supervisor**
E. G. Todd	*A. Prof E. Kemp*
Institute of Information Sciences and Technology	*Institute of Information Sciences and Technology*
AH3.84	*AH3.87*
Extension: 2455	*Extension: 2469*
Email: *e.todd@massey.ac.nz*	Email: e.kemp@massey.ac.nz

About UI Patterns

If you are interested in finding out more about UI Patterns visit the following Web Sites:
Yahoo Design Pattern Library - http://developer.yahoo.com/ypatterns/index.php
Tidwell's Design Interface patterns - http://time-tripper.com/uipatterns/
Welie's patterns in Interaction Design - http://www.welie.com/patterns/

Note

This project has been evaluated by peer review and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees. The researcher(s) named above are responsible for the ethical conduct of this research.

If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher(s), please contact Professor Sylvia Rumball, Assistant to the Vice-Chancellor (Ethics & Equity), telephone 063505249, email:humanethics@massey.ac.nz.

A4.2 Study Two – Information Sheet

Using UI Patterns to Guide the Development of Conceptual User Interface Models

INFORMATION SHEET

All students are required to complete the exercises that make up this tutorial as part of the course requirements. Only responses from those students who agree to participate in the experiment will be used for this research. After filling in the consent forms, the forms will be collected and placed into a sealed envelope so the researcher and the research assistant who are taking this tutorial session will have no knowledge of who has or has not agreed to have their results included in the research.

Introduction

The overall purpose of this research is to investigate whether a user interface pattern language can be used to successfully guide novice UI developers in the creation of conceptual user interface models. The investigation is aimed at evaluating a method for introducing students to UI patterns with the aim of developing their understanding of pattern languages and the user interface knowledge contained within the patterns and then applying some of that knowledge. .

The goals of this experiment are:

- To collect information about a suitable pattern format for UI patterns to use with student UI designers.
- To evaluate whether the UI-pattern modelling method from TUIPL can successfully guide students in creating a UI-pattern model.
- To evaluate a method for using the content of the UI patterns to guide the development of a UI conceptual model.

This research is being undertaken as part of doctoral studies by the researcher Elisabeth Todd. Supervision of this project is by Associate Professor Elizabeth Kemp and Associate Professor Chris Phillips of the School of Engineering and Advanced Technology.

Participant Recruitment

The activities necessary for this research will be conducted as part of the course for 158.359 - Human-Computer Interaction. Students enrolled in this paper are invited to have the analysis of their work included as part of the research.

The staff involved in teaching 158.359 will <u>not</u> know which students have agreed to allow their data to be included in the analysis of the experimental results. No student will be penalized for not agreeing to their data being included.

Project Procedures

The course lecturer will have no knowledge of who has agreed to have their results used in the research. Students' work will be marked by the course lecturer who is using assessment criteria independent to those used by the researcher. A copy of the student's work will then be made and passed to the researcher.

Only the researcher will collate the research data. Once the data from individual students has been matched each set will be given an identifier then the cover sheets identifying individuals will be removed and destroyed. Data collected from individuals who have not consented to having their data included will be destroyed. The data collected will be held for the duration of the research and will be kept in a secure location and will be destroyed when the research is complete.

The data for this project will be collected in the following ways:

- Photographs of the work surfaces taken at timed intervals – All features that could be used to identify an individual will be removed;

- The written results of the modelling exercises will be analysed for the way the models have been constructed;
- A Questionnaire about the set of patterns.
- A Questionnaire covering canonical abstract prototyping components used to create the conceptual UI models.
- A Questionnaire about the methods, patterns and conceptual UI modelling.

The results from the analysis of the experimental data will be available from the CS office, AH3.91 on request. Students will have to present their ID card to prove that they were enrolled in this class and show they have a right to the information.

Participant involvement

As a participant in the experiment you will follow the same procedure for the tutorial as the other members of the class. The plan for the tutorial is as follows:

1. Students to choose working partner.
2. Introduction to explain the details of the study.
3. Complete ethics approval forms - All approval forms placed in a sealed envelope.
4. Presentation on creating a UI-pattern model.
5. Answer conceptual UI modelling questions.
6. Complete Exercise One.
7. Fill in patterns questionnaire.
8. Have a short break.
9. Presentation on creating a UI conceptual model.
10. Complete Exercise Two.
11. Answer canonical abstract prototype questions.
12. Complete the exit questionnaire.

Participant's Rights

You are under no obligation to accept this invitation. If you decide to participate, you have the right to:

- Ask any questions about the research at any time during participation;
- Decline to answer any particular question related to the research;
- Decline to have your work surface photographed;
- Withdraw from the research before the end of the tutorial;
- Provide information on the understanding that all identifying information will be removed so that it becomes anonymous;
- Have access to a summary of the project findings when it is concluded.

Project Contacts

Researcher	Supervisor
E. G. Todd	*A. Prof E. Kemp*
School of Engineering and Advanced Technology	*School of Engineering and Advanced Technology*
AH3.84	*AH3.87*
Extension: 2455	*Extension: 2469*
Email: *e.todd@massey.ac.nz*	Email: e.kemp@massey.ac.nz

About UI Patterns

If you are interested in finding out more about UI Patterns visit the following Web Sites:
Yahoo Design Pattern Library - http://developer.yahoo.com/ypatterns/index.php
Tidwell's Design Interface patterns - http://time-tripper.com/uipatterns/
Welie's patterns in Interaction Design - http://www.welie.com/patterns/

Note

This project has been evaluated by peer review and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees. The researcher(s) named above are responsible for the ethical conduct of this research.

If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher(s), please contact Professor Sylvia Rumball, Assistant to the Vice-Chancellor (Ethics & Equity), telephone 063505249, email humanethics@massey.ac.nz.

A4.3 Study Three – Information Sheet

Developing of Conceptual User Interface Models Guided by UI Patterns

INFORMATION SHEET

This is the background information to the UI design exercises you have been asked to participate in. If you agree to take part you need to be able to commit between three to five hours of your time. You will be asked to complete a consent form. These forms will be collected and kept in a safe place for five years.

Introduction

The overall purpose of this research is to investigate whether a user interface pattern language can be used to successfully guide UI developers in the creation of conceptual user interface models. The method is specifically aimed at novice UI developers but should also be helpful when a development team includes members of the user community.

This investigation follows on from two previous sets of experiments. The first focused on the evaluating a method for introducing students to UI patterns with the aim of developing their understanding of patterns and pattern languages. It also investigated the impact of different pattern content presentations on student acceptance and use. The second experiment applied findings from the first but focused on whether students could learn new user interface knowledge contained within the patterns and then apply that knowledge to create suitable conceptual models for a user interface.

The goals of this investigation are:

- To evaluate the first two methods in the TUIPL framework to develop conceptual UI models.
- To collect information about how the methods could be improved.
- To collect opinions on the viability of the first two methods in the TUIPL framework in a professional setting

This research is being undertaken as part of doctoral studies by the researcher Elisabeth Todd. Supervision of this project is by Associate Professor Elizabeth Kemp and Associate Professor Chris Phillips of the School of Engineering and Advanced Technology.

Participant Recruitment

The participants invited to take part in this investigation have a range of experience in the development user interfaces. They all have associations with Massey University and most are currently employed as lecturers or in computer support, or are studying towards higher degrees. Some have industrial IT experience.

Participant involvement

Potential participants who on reading this information sheet agree to take part in the research will be asked to complete an ethics consent form and a set of questions about their user interface development experience. They will be given a set of UI patterns to study. A meeting time for the first two-to-three hour design session will be organised.

At the first design meeting the tasks to be carried out are:

1. Study the introductory material and with the researcher work through an example of the UI-pattern modelling method.
2. Study the initial requirements for the exercise example then apply the UI-pattern modelling method to create a UI-pattern model for this example.
3. Next the participant will complete a questionnaire, discuss any problems they identify with the UI-pattern modelling method and suggest improvements.
4. A meeting time for the second one-to-two hour design session will be organised.

Page Break

Page 1 of 2

At the second design meeting the tasks to be carried out are:

1. Review the initial requirements and the UI-pattern model for the worked example then with the research work through the example of the TUIC method.
2. Review the initial requirements for the exercise and study the UI-pattern model exemplar. Apply the suggested method and complete a TUIC model for the UI using canonical abstract prototyping and navigation components.
3. At the end of the session there is an exit questionnaire to be completed and a debriefing discussion.

Data collection

- Discussion to identify any problems with the methods and suggest improvements, at stages throughout the process.
- Observations by the researcher as the modelling exercises progress.
- The results of the modelling exercises will be analysed to appraise the way the models have been constructed;
- A set of Questionnaires providing background information and asking about your experience with: the proposed methods, patterns, UI-pattern modelling and abstract prototyping.

About UI Patterns

If you are interested in finding out more about UI Patterns visit the following Web Sites:

- Yahoo Design Pattern Library - http://developer.yahoo.com/ypatterns/index.php
- Tidwell's Design Interface patterns - http://time-tripper.com/uipatterns/
- Welie's patterns in Interaction Design - http://www.welie.com/patterns/

Participant's Rights

You are under no obligation to accept this invitation. If you decide to participate, you have the right to:

- Ask any questions about the research at any time during participation;
- Decline to answer any particular question related to the research;
- Withdraw from the research before the end of the tutorial;
- Provide information on the understanding that all identifying information will be removed so that it becomes anonymous;
- Have access to a summary of the project findings when it is concluded.

Project Contacts

Researcher	**Supervisor**
E. G. Todd	*A. Prof E. Kemp*
School of Engineering and Advanced Technology	*School of Engineering and Advanced Technology*
AH3.84	*AH3.87*
Extension: 2455	*Extension: 2469*
Email: *e.todd@massey.ac.nz*	Email: e.kemp@massey.ac.nz

Note

This project has been evaluated and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees but has been registered into the low risk data base. The researcher(s) named above are responsible for the ethical conduct of this research.

If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher(s), please contact Professor Sylvia Rumball, Assistant to the Vice-Chancellor (Ethics & Equity), telephone 063505249, email humanethics@massey.ac.nz.

Appendix A5: Tutorial Workbook

Conceptual Modelling Exercises

A5.1 UI Pattern-Informed UI Design

This manual provides a worked example applying the early stages of a user interface design informed by a set of UI patterns. The **T**eaching **UI** design guided by a **P**attern **L**anguage framework (TUIPL).

2003).

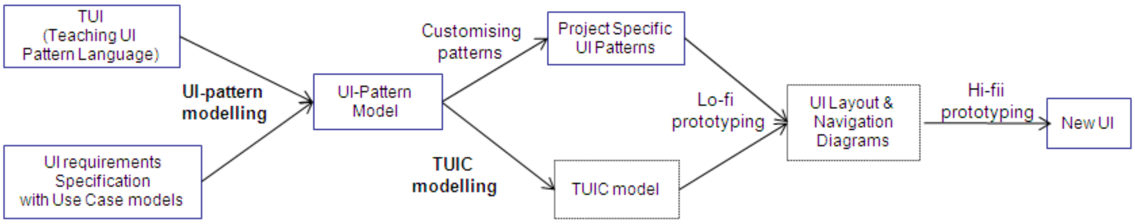


Figure A5.1 - Over view of the TUIPL framework

This framework (Figure A5.3) assumes that an initial user interface requirement document exists. The TUIPL framework has been developed with the goal of teaching UI design; specifically for helping students develop conceptual models for user interfaces. Alternatively the TUIPL framework could be applied by a design team which includes user group representatives. A conceptual UI model provides an abstract description of a UI. The **T**eaching **UI C**onceptual (TUIC) modelling method uses canonical abstract (CAP) prototype components and navigation components. Two papers introducing these components have been included in the appendices (Constantine 1998, Constantine *et al.* 2003).

The framework process starts with the development of a UI-pattern model. Initially the UI requirements are examined and the required actions and presentation spaces are identified. Next, suitable patterns are selected from the pattern languages and a UI-pattern model is created. The essential features of the solution to the problem each pattern addresses are represented by a TUIC model diagram. The second stage is to use these TUIC model diagrams as a guide to develop a TUIC model for the UI. Finally the TUIC model can be restructured into associated layout and navigation models.

The patterns used are interaction design patterns (Dearden & Finlay, 2006) which illustrate the issues underpinning user interface design decisions. Patterns provide the reader with a known solution to a recognised problem in a structured format. As well as the rationale behind the solution they describe how the solution should be applied including tradeoffs that need to be considered. The accompanying TUIC model diagram highlights the essential elements of the solution. Patterns act as a *lingua* *franca*, that is, a communication device and a means of illustration when discussing interface design (Bayle et al., 1997; Erickson, 2000; Dearden & Finlay, 2006). When linked together in a pattern language, relationships between the patterns should be synergistic, in that the language should provide an understanding of UI design that is greater than just the application of individual patterns.

UI Patterns

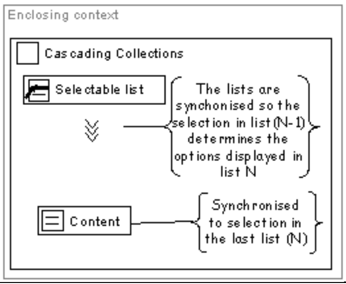
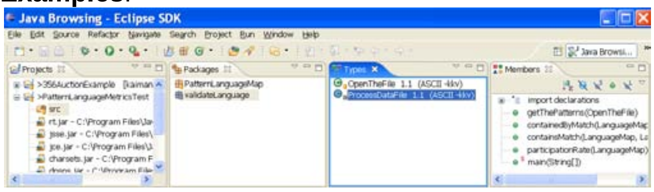
Front	Back
8 - Cascading Collections **Context** The content to be accessed… **Problem** How should series …? **Solution** *Express a hierarchy by ….*  	**Reference** This pattern effectively … **Forces** …. **Discussion** … **Examples**:  Eclipse showing the Java Browsing view: projects>packages>types>members …

Figure A5.2 – Formatting of sections making up the patterns used for this exercise

The exercise uses a restricted set of just nineteen patterns (see appendices). These patterns have been structured for teaching students. They are based on those created by Tidwell (1999, 2006) and describe standard UI contexts. They link together into a consistent pattern language structure. The set is small, so design choices are limited as is the set of potential UIs that can be described.

Each pattern used in this modelling exercise is restricted to one double-sided A4 page and contains sections of information as shown in Figure A5.2. The essential parts of these patterns are the context, the problem and the solution. The context can be thought of as ‘applicability’ characterising situations where the pattern can be applied. Normally the context is defined in terms of high-level patterns that use the pattern. The problem section defines the design issues or situation for which the pattern provides a solution. The solution section addresses the problem and may be in the form of a set of instructions. Associated with the solution is a TUIC model diagram that identifies the essential aspects of the solution.

This worked example provides guidance in developing two types of UI conceptual model, a UI-pattern model and a TUIC model.

A5.2 Case Study

This is an incomplete solution developed to illustrate the proposed pattern-driven UI design method.

INITIAL REQUIREMENTS DOCUMENTATION

The design method assumes that an initial requirements document has been completed and that conceptual or abstract modelling of the UI is the next stage. While you study this example remember the methods are intended to be applied by a UI design team of either students or a combination of professional and non-UI professionals.

Narrative Overview – Course Registration System

At the beginning of each semester, students at Eastern State University must register for a programme of study consisting of between one and five courses. A new system is to be implemented which will support on-line registration. The system must meet the following requirements:

Requirements

Functional Requirements

- Provide students with details of courses on offer in that semester.
- Assist students in assembling a programme of study, by allowing them to add and delete courses until they are satisfied with their programme.
- Maintain a running total of the cost of the programme.
- Carry out checks to make sure that students are eligible to enrol for the courses they wish to add, and that they do not enrol for more than five courses per semester.
- Allow students to submit their programme.
- Allow students to check their enrolment history (details of courses studied previously and grades obtained).
- ....

Non-Functional Requirements

Students use this system during two short periods each year, and some students will be new to the system. It is therefore important that the system is easy and intuitive to use, and that in particular it should:

- Present information in a clear, well organised and concise manner.
- Provide good support and feedback at each stage.
- Support an efficient and flexible dialogue, including a good search capability for courses.
- Be satisfying to use.
- ....

Scenarios

The following scenario is representative of the use of the system by a student returning for a third year of study:

Third year student

Dan wishes to construct his course of study for the first semester of his third year. He enters his student ID and password, and the system responds by displaying his identifying details (student name, student number, degree and majoring subject). The system offers access to details of courses being offered by the university in that semester (course number, course title, prescription, prerequisites, lecturer(s), points-value and cost). The system also provides Dan with the option of reviewing his academic record to date. This is useful because he may be unsure as to which courses he has already passed, and exactly how many courses he needs to complete his degree. Dan can add courses to his programme. The system may refuse to allow Dan to enrol for a course, if for example he does not have the appropriate prerequisite(s). Dan can delete previously selected courses from his programme, or alternatively chose to delete his entire current programme. As Dan adds or deletes courses, the system updates the programme, displaying the current courses he has selected. The display also shows the cost of each course plus the current total cost. At any time, Dan can submit his programme to the system, which records the programme and displays a confirmation. He can also quit the session at any stage but if his current programme contains un-submitted courses a warning message will alert him and allow him to submit them as he quits.

Other scenarios …

Structured User Role Model

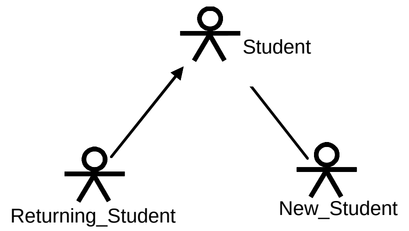


Figure A5.3 - Role map showing returning and new students as types of student

Returning Student

Goal:	to re-register with the university for a semester programme of up to five courses	
Background:	* educated to 7^{th} form or equivalent * has an existing academic record (enrolment history) * computer literate	
Domain Knowledge:	* will have enrolled previously * has some knowledge of courses and programmes	
System:	will have some prior knowledge of enrolment procedures	
Proficiency:	novice to intermediate	
Interaction:	* intermittent but intensive (uses the system frequently over short time-frames twice per year) * user driven	
Information:	* flows both ways * information out is large and complex * information in is low in volume	
Functions:	no special functions	
Usability:	* good quality presentation of information * good support and feedback	* efficient, flexible and reliable * satisfying

Table A5.1- Details of role 'Returning Student' which specialises Student

Other roles …

Task Model

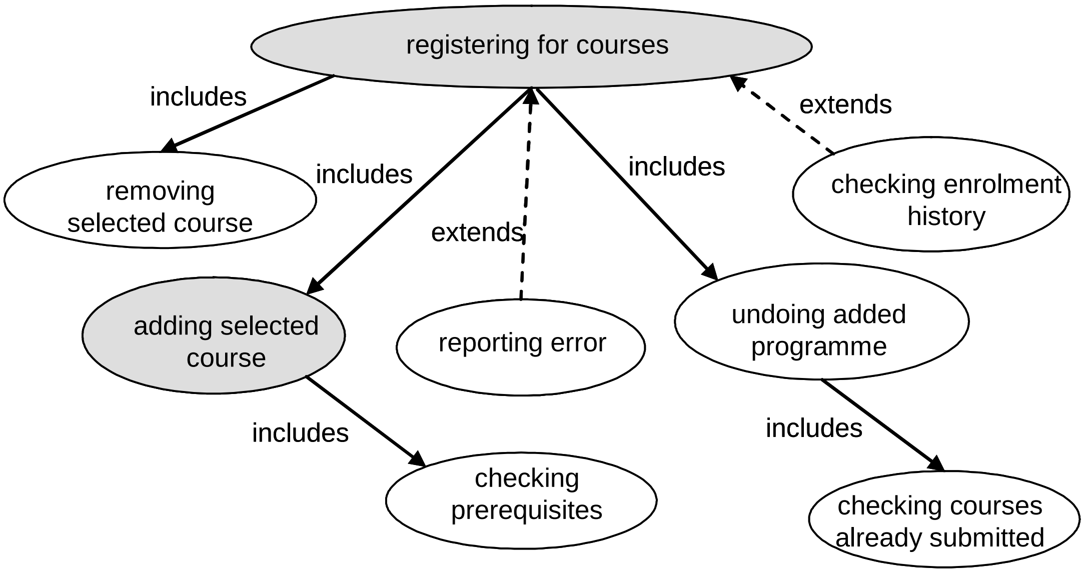


Figure A5.4 - Use case map with those shaded to be modelled

Use case details

Once a use case is completed it is examined to identify actions and interaction spaces. These are itemised in the accompanying table. The notifications and error reporting are included in these use cases but will not be included in be modelling exercises.

Registering for courses

In order to register for a programme of study, a student must submit a selection of between one and five courses

ROLES SUPPORTED: returning student, new student

PRE-CONDITIONS: student logged in, system initialised

USER INTENTIONS	SYSTEM RESPONSIBILITIES
optionally at any point do: checking enrolment history	**optionally at any point do:** reporting error
	1. present student's details 2. set default for finding a course by pre-selecting current majoring subject 3. offer list of associated courses (synchronised to majoring subject) 4. pre-select first course in list and present its details
5. **optionally:** [select a subject]	
	6. change course list to that associated with selected subject 7. present first course in list's details in space for finding a course
8. **optionally**: [select a course]	
	9. present selected course's details in space for finding a course
10. **optionally do:** adding selected course; removing selected course; undoing added programme	
11 **optionally:** [specify that current programme is to be submitted]	
	12 update record of coming semester's current programme in the student's study programme space 13 notify user of all courses accepted (student can submit courses one or more at a time so system keeps track of which courses have already been submitted)
14 **optionally**: [specify that all courses added to the coming semester's	

current programme are to be deleted]	
	15 **do**: <u>undoing added programme</u>
16. **optionally**: [specify that student's study programme is to be printed]	
	17: format student's study programme for printer and send, notifying users of progress
18. quit the session	
	19. check all added courses have been submitted if not notify user and provide option to submit
	20. end session processes

POST-CONDITIONS: notify authorities of new or changed enrolment

Use Case "registering for courses"

Actions	Interaction spaces
	Find a course
Select subject	➢ List of subjects
Select course	➢ List of courses for selected subject
	➢ Course details for selected course
Adding selected course to programme Removing selected course from programme Undo added programme, all courses	
	Study programme
Submit programme to enrolment authority Delete programme already submitted Print current programme or enrolment history	➢ Current programme of courses added for the coming semester ➢ Student details ➢ Enrolment history
Checking enrolment history Returning to semester's current programme	Error reports
	Notify submission progress and success Notify printing progress Notify if courses not submitted when quitting session
Quit current enrolment system	

Table A5.2 - Inventory of actions & spaces for use case "register for courses".

Adding selected course

Adds a selected course into a student's programme

ROLES SUPPORTED: returning student, new student
PRE-CONDITIONS: course is selected

USER INTENTIONS	SYSTEM RESPONSIBILITIES
1. specify that selected course is to be added to current programme	
	2. **do**: checking prerequisites
	3. **either** add course to programme and update cost, if prerequisites satisfied, **or** notify user of problem

POST-CONDITIONS: course added to programme, if prerequisites satisfied

Other use cases …

Details of Interaction Spaces

Each use case is examined to identify actions and interaction spaces. This information has been included for completeness but is not required to complete the modelling exercises.

Interaction Spaces	**Content**	
List of subjects	Subject name	
List of courses	Course code	Course name
Course details	Course code Course name Course description	Course controller Number of points Semesters available Cost
Student details	Student name Major subject	ID number
Current programme	Course Code Cost	Course name Running total of costs
Enrolment history	Year taken	…
Error reports	As appropriate	….
Notifications dialogues …		…

Table A5.3 - Interaction space details for use case “registering for courses.

The example UI-pattern model

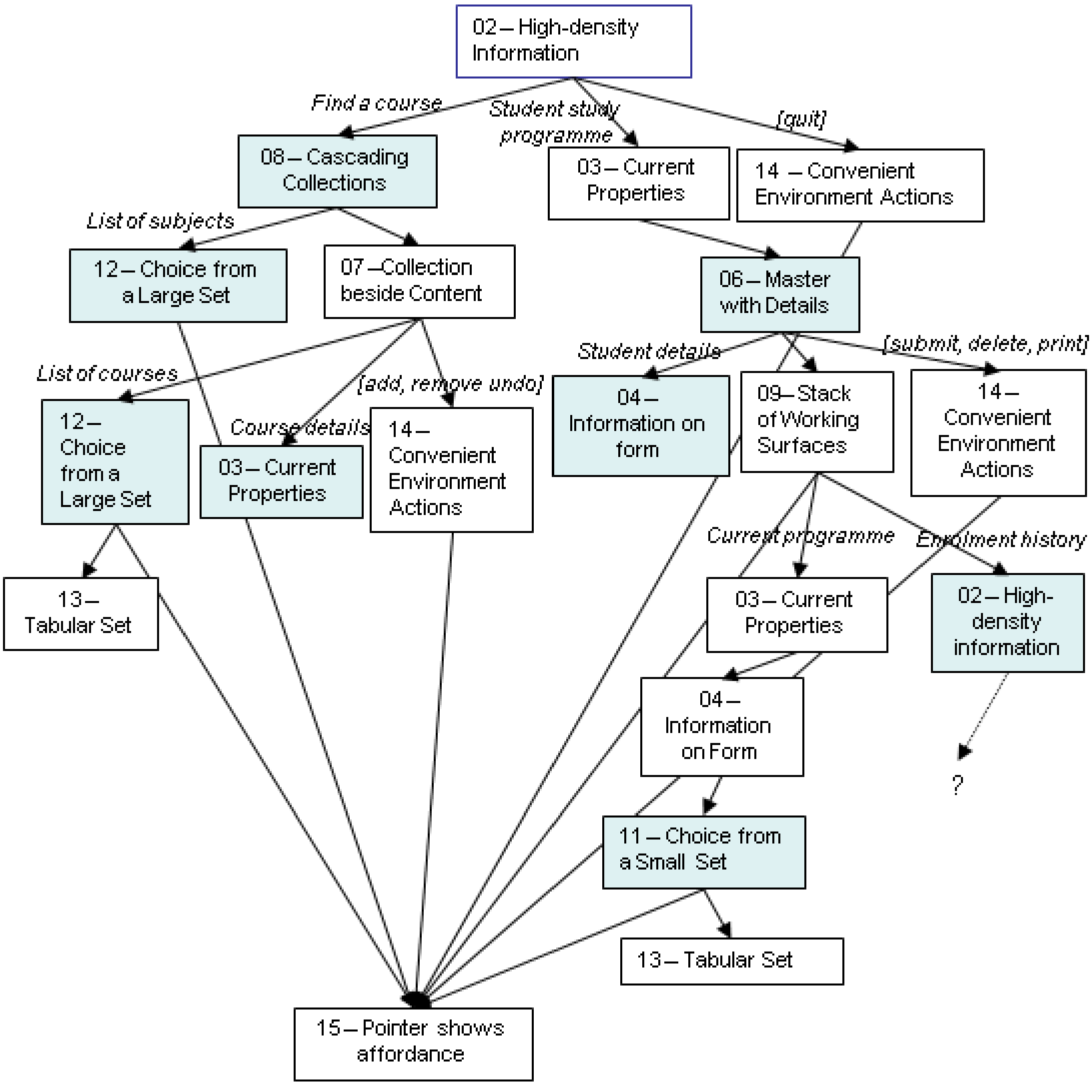


Figure A5.5 - UI-pattern model for the course registration system

A5.3 UI-PATTERN MODELLING

UI-pattern modelling starts with an examination of the UI requirements to identify the required actions and interaction spaces, if these have not been provided. A UI-pattern model is a selection of patterns that are structured into a strongly hierarchical network describing the proposed user interface. For ease of the following discussion, a hierarchical or tree-like terminology will be used.

Using the list of actions and interaction spaces as a guide, the pattern language is examined and one or more patterns selected, for example describing one of the interaction spaces. Identifying suitable patterns then continues using the context section in each selected pattern to build up towards a root pattern for the UI-pattern model. The root pattern provides an overall description of the UI. The root pattern will be one of the lower numbered patterns.

UI details are described by higher numbered patterns lower in the hierarchical structure of the UI-pattern model. These are selected by consulting the reference section of each previously selected pattern, until no further low-level patterns are required. Although the description of the method is sequential, in practice it is expected that the selection of patterns and building the UI-pattern model will be cyclic as the designer chooses and rejects patterns and restructures the UI-pattern model until a suitable structure has been created.

The method

The main steps for creating a UI-pattern model are shown in Figure A5.6 with details in Table A5.4. This method takes an essentially middle-out approach to development.

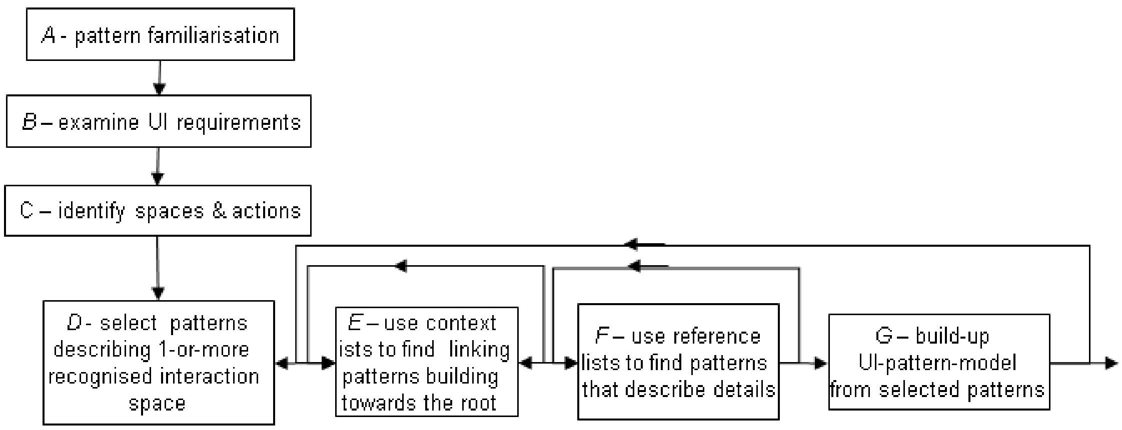


Figure A5.6- Schematic of the UI-pattern model building method

Step A.	Become familiar with the UI patterns.
Step B.	Examine the UI requirements documentation.
Step C.	Identify actions and interaction spaces then group related interaction spaces and any associated actions.
Step D.	Select patterns that seem to best describe one or more of the interactions spaces.
Step E.	Check each pattern's Context section and select a pattern from the list that gives the best overall description. This description may include patterns already selected. A pattern may be mentioned in the context section of one or more of the selected patterns. Use the context section to find common patterns until one of these higher level patterns can be identified as the root (1, 2 or 3) of the UI-pattern model
Step F.	Check each pattern's Reference section to identify patterns that best describe the details and additional features required for the user interface.
Step G.	Using selected patterns, create a diagram consisting of patterns connected with lines to indicate how they are linked into a structure. Use the context and reference sections of patterns to check for missing links or patterns.

Table A5.4 - Method for creating a UI-pattern model.

The first step is to become familiar with the set of UI patterns. The pattern languages used for this workbook example can be found in the appendices. Obviously over time this step would not be as necessary because most of a design team would already be familiar with the UI patterns.

The designer also needs to become familiar with the requirements for the new user interface.

The extent of step C depends on whether the interaction spaces and actions have been identified as part of the initial requirements documentation. Once they exist, as in this case study (see copy of the table below) the spaces and actions should be organised so that related ones are in groups. In this example, the list of subjects, list of courses and course details spaces together make up the Find a course space along with the actions of selecting the subject or selecting the course. Adding or deleting courses relate the course list of the Find course space with the current programme in the Student's study space as does undoing the whole programme. In the student's study space, once the current programme has been created for the specified student then it can be submitted, deleted or printed. The student's enrolment history can be viewed from the students study space too. Finally the system can be exited via the quit action. These groupings can be used to guide steps D, E and F.

Actions	Interaction spaces
	Find a course
Select subject	List of subjects
Select course	List of courses for selected subject
	Course details for selected course
Adding selected course to programme	
Removing selected course from programme	
Undo added programme, all courses	
	Study programme
Submit programme to enrolment authority	Current programme of courses added for the coming semester
Delete programme already submitted	
Print current programme or enrolment history	Student details
Checking enrolment history	Enrolment history
Returning to semester's current programme	
	Error reports
	Notify submission progress and success
	Notify printing progress
Quit current enrolment system	Notify if courses not submitted when quitting session

Figure A5.7 – Duplicate - Inventory of actions & spaces for use case "register for courses".

Building the UI-pattern model

Start step D by selecting one of the groups of related interaction spaces (Table A5.2) and then locate a couple of patterns that clearly describe one of the interaction spaces. For example in this case study the problem of displaying the interaction space 'List of subjects' is best represented by Pattern '12 Choice from a large set' which is applied in the context that "*The artifact shows, or allows the user to select or set, a value which is one out of a large set of possible values (normally more than ten) where it is either inappropriate or not possible to display all values in the list at once.*" as shown in Table A5.5. Similarly 'List of courses' can be represented by this pattern too.

Interaction space	Pattern	Part of the pattern's context section
List of subjects List of courses	12 Choice from a large set	The artifact shows, or allows the user to select or set, a value which is one out of a large set of possible values (normally more than ten) where it is either inappropriate or not possible to display all values in the list at once. …
Course details	03 Current properties	The information describing an artefact must be displayed. The information maybe changed over time. …

Table A5.5 - Matching interaction spaces to possible UI patterns

The interaction space 'Course details' provides additional information about any course selected so requires a simple solution. A couple of patterns from this pattern language might be suitable: '03 Current Properties' and '04 Information on Form'. Pattern 4 is used when "*A user often needs to view information that provides a snapshot of something represented by values as in CURRENT PROPERTIES*". An examination of the content of course details in Table A5.3 shows that the content is reasonable homogeneous and represents a course so this pattern has potential. The pattern '03 Current Properties' is one of the high level patterns and is therefore comparatively general, describing many of situations. The context for this pattern starts with the statement "*The information describing an artefact must be displayed. The information maybe changed over time.*" This context certainly covers displaying the course information which may also change over time, for example when the course controller is replaced by a new lecturer. As the context of Pattern 4 mentions that it is applied in the context of Pattern 3, at this point in the modelling exercise select the pattern higher up the hierarchy. Both the patterns (12 & 3) have more detail in their context sections which should also be considered. When selecting a pattern, at a minimum, the problem and solution sections should be examined.

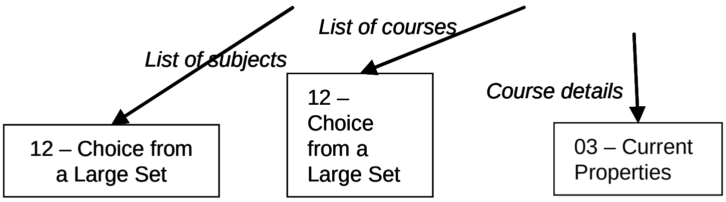


Figure A5.8 - The initial patterns select to represent the collections of subjects and courses

Figure A5.8 shows the three patterns representing the list of subjects, the list of courses and the details of any course selected in the list of courses. Arcs are shown linking into the nodes. The arcs are labelled with the interaction space the linked node and any subsequent 'sub-tree' represents.

Linking the patterns

The next step E links some of the selected patterns through common higher-level patterns. For example when checking Table A5.2 the space '<u>Course details</u>' is identified with the phrase '*Course details for selected course*' indicating there is a link or common parent between this space and the '<u>List of courses</u>'.

Interaction space	Pattern	Part of the pattern's context section
List of courses	12 Choice from a large set	… It may also be used in conjunction with more complex list like organisation as found in **COLLECTION BESIDE CONTENT** and CASCADING COLLECTIONS …
Course details	03 Current properties	… this pattern can also be used for the content part of the pattern **COLLECTION BESIDE CONTENT** or for information on a surface in a STACK OF WORKING SURFACES and also …
	07 Collection beside Content	A collection of objects, categories, or even actions needs to be displayed and each of these items has content associated with it that the user wants to access. The user needs to see the overall structure of the collection so they can select items of interest in any order. …
	14 Convenient Environment Actions	… This pattern is also required when actions need to be carried out on the selected content displayed using **COLLECTION BESIDE CONTENT**.

Table A5.6 - Locating higher level patterns

The individual patterns along with 'The Patterns List' can be used to identify patterns that can form the context for a specific pattern. The context list for pattern '03 Current Properties' representing course details is [02, 07, 10] and for pattern '12 Choice from a large set' representing the list of courses is [02, 04, 05, 07, 08]. The common patterns from these lists are [02, 07]. Further examination of the context section of the two patterns 12 and 3 (see Table A5.6 for the relevant extracts) indicates that the pattern '07 Collection beside Content' is the most appropriate common parent for this model. This is confirmed by examining the problem section of Pattern 6 which says "*How can you present a large amount of content matched to each member of a set of items so that a user can explore it at their own pace, in a way which is comprehensible and engaging to the user?*" The 'list of courses' space matches the set of items or the collection, while the 'Course details' space matches the content. The context section of Pattern 6 also confirms that it links members of a collection or list with associated details.

Next the reference section of Pattern 6 is used to confirm the link. Table A5.7 reproduces part of the reference section showing that it includes the phrase "*a form of select and show"* which clearly describes the course list and course details pairing.

Pattern	Part of the pattern's reference section
07 Collection beside Content	This patterns is basically a form of select and show … For the content section use one of the high level patterns such as Narrative OR CURRENT PROPERTIES that best represents the overall structure of the content being displayed. Then where necessary select related lower level patterns to fully represent the structure of the content. Where the user can carry out actions on the selected item provide these as a set of **CONVENIENT ENVIRONMENT ACTIONS.**

Table A5.7 - Locating lower level patterns

Illustrating the method indicates that the designer moves between checking context sections, step E and checking reference sections, step F. The reference section is checked to identify any other patterns that should be considered. For example, to work with the two patterns 12 and 03 already selected. Pattern '07 Collection beside Content' also identifies patterns [01, 10, 11, 14] as candidates for implementing it. The Pattern '14 Convenient environment actions' should be considered for any actions that could be associated with this pattern. For example, from the groupings of interaction spaces and actions it is clear that students need to add a selected course to their current programme. The actions add, remove and undo can be represented by Pattern 14.

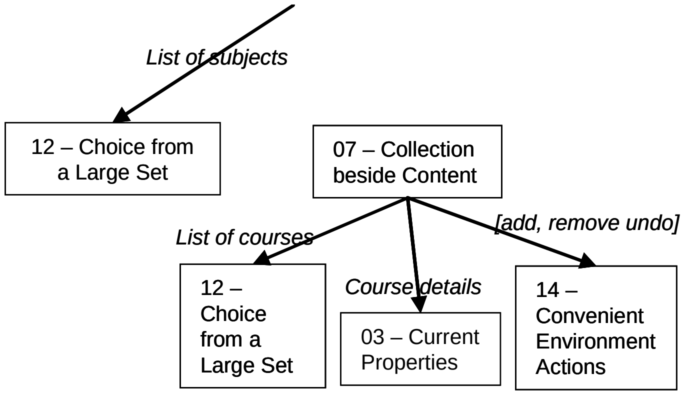


Figure A5.9 – Part of the UI-pattern model for the two lists

By applying step G the UI-pattern model begins to take shape as shown in Figure A5.9. The process continues by trying to represent the relationship between the 'list of subjects' and the 'list of courses' associated with each subject. Steps E, F and G are repeated. The list of patterns that can form the context for pattern 12 is [02, 04, 05, 07, 08] and for pattern 07 is [01, 02, 06, 08] with the common patterns being [02, 08]. An examination of Pattern '<u>08 Cascading collections</u>' shows that it is a good description of two lists and the associated course details which together will be used to find courses for a student's current program.

The example UI-pattern model

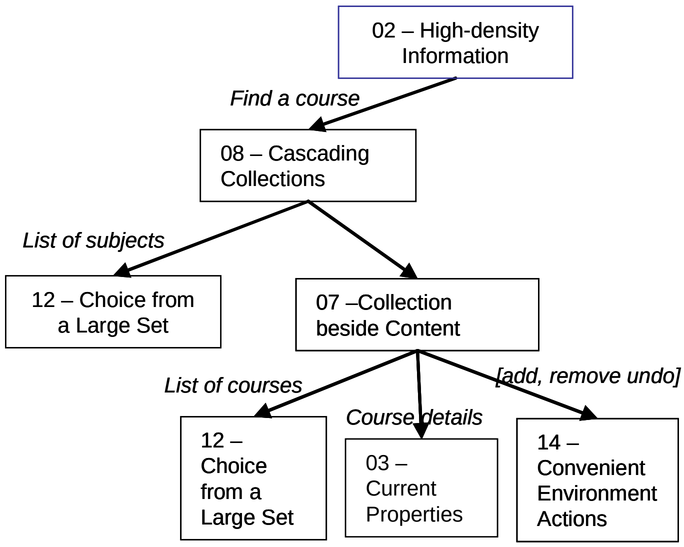


Figure A5.10 - Part of the sub-tree representing the interaction spaces required to find a course

One pattern becomes the root providing an overview of the UI being developed. In this example the context section of Pattern '<u>07 Collection beside content'</u> identified pattern '<u>08 Cascading collections</u>' as a good description of the dependent interaction spaces:

'List of subjects', 'List of courses' and 'Course details'. This sub-tree of patterns represents the interaction space for finding a course.

Moving up a level the context section of Pattern 8 identifies Pattern '02 High-density information' as the root pattern resulting in the partially completed tree seen in Figure A5.10.

At this point a similar process can be used to model the interaction spaces making up a student's study programme. Alternatively, a top-down approach can be taken. The lower levels are found by following patterns listed in the reference sections. Often more than one of the members from the reference list of potential patterns is required to complete a pattern.

As can be seen in Figure A5.5 three patterns are required to represent the first level below the root pattern. The Pattern '08 Cascading collections' represents interaction spaces for finding a courses as a student searches to locate one they may enrol in. The Pattern '03 Current properties' via pattern '06 Master with details' groups the three interaction spaces representing a student's programme of study made up of student details along with their enrolment history and the current programme they are constructing. Finally, the Pattern '14 Convenient environment actions' is required for the action to quit the system.

An examination of the model (Figure A5.5) shows that some patterns are used repeatedly. It is up to the modeller to decide whether to use multiple copies of a pattern or to use multiple arrows pointing to a single copy. In this model copies have been used in most cases because each copy represents a different element or group of elements in the UI. For the Pattern '15 Pointer shows affordance' only one copy has been used because this action is similar in each of the interaction spaces.

As with all models the modeller may choose to show only links and patterns that they deem necessary. For example, 'Course details' is represented by just pattern 3 but extending this into a sub-tree including Pattern 4 could be considered. Another example is where the modeller may choose to exclude most links to Pattern 15 so the diagram does not become cluttered. They would then append an annotation to indicate that Pattern 15 should be implemented for all groups of environment actions.

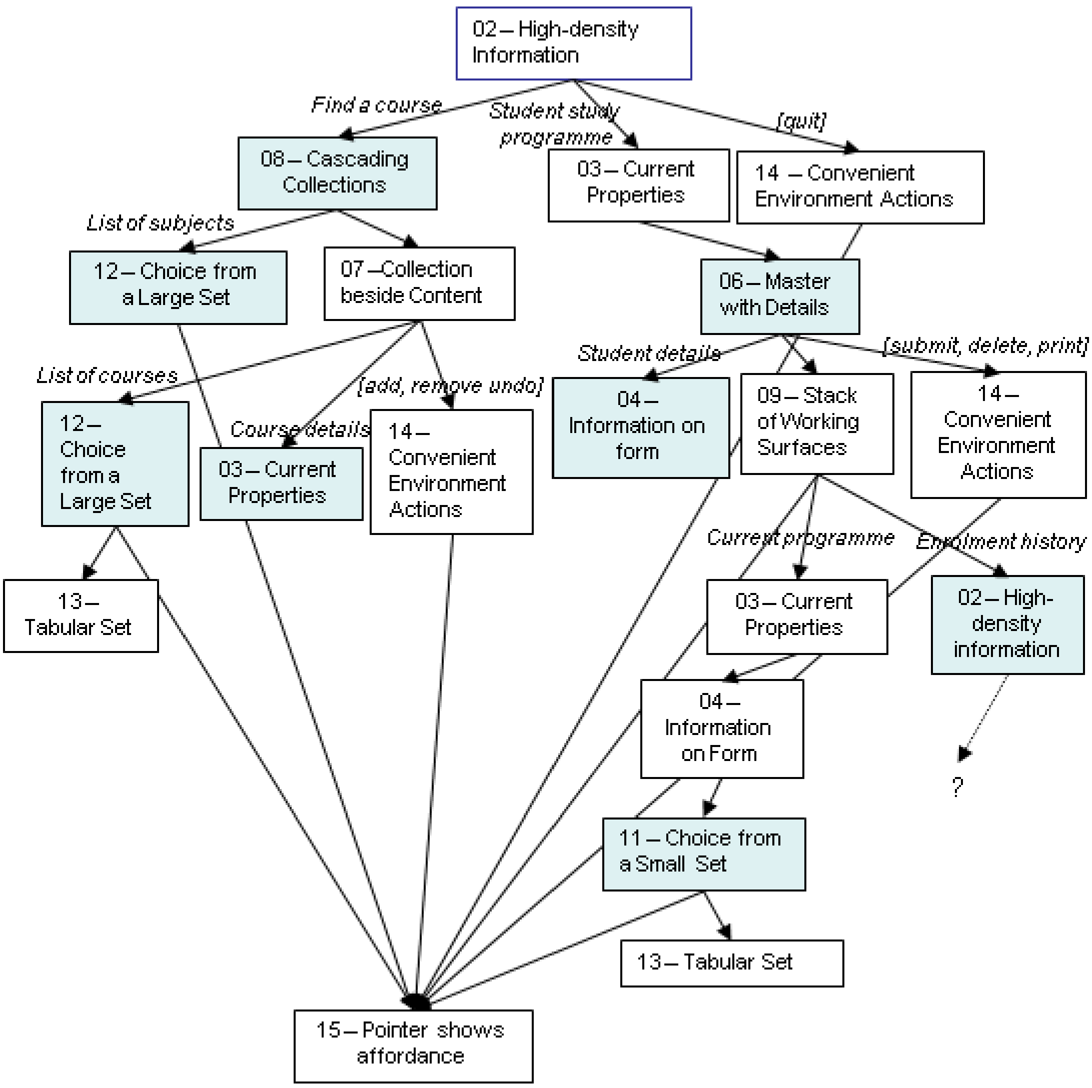


Figure A5.11 - UI-pattern model for the course registration system

The sub-tree representing the student's enrolment history has not been included in this example hence the question mark under the high level pattern describing this information.

Interaction spaces	Potential pattern
List of subjects	12 CHOICE FROM A LARGE SET
List of courses	13 TABULAR SET
Course details	03 CURRENT PROPERTIES
Student details	03 CURRECT PROPERTIES
Current Programme	13 TABULAR SET
Enrolment History	2 HIGH DENSITY INFORMATION

Table A5.8 – Interaction spaces matched to patterns.

Table A5.8 shows the match between the interaction spaces (shown shaded in Figure A5.5) and the lower level patterns representing them. The root pattern for each sub-tree representing an interaction space may be at a higher level. The linked higher level patterns indicate which spaces are logically grouped together and lower level patterns add additional details.

--------→> PLEASE STOP HERE TO DISCUSS ANY ISSUES OR IDEAS YOU MAY HAVE ABOUT THE METHOD AND THE UI-PATTERN MODEL <←-----

A5.4 TO DO

Please study the accompanying exercise example requirements and complete a partial UI-pattern model from the requirements.

A5.5 TUIC MODELLING

Once the UI-pattern model has been created, a TUIC model for the UI is developed. The essential elements of each pattern's solution are illustrated using a TUIC model. These generalised examples are used to guide the development of the TUIC model. In many systems this will result in a series of TUIC model diagrams although the example created in this workbook results in a single 'screen'.

The main interaction space is used to contain related spaces making up the system. Either the details of each section of the interface or in some models the links to sections are then added to this space. At first generic CAP symbols can be used to represent different levels of detail but as modelling progresses it should become clearer where specific symbols can be substituted. Although pattern names may be used initially, in the completed model naming of the different areas should reflect the application being developed and be meaningful within that context.

The method

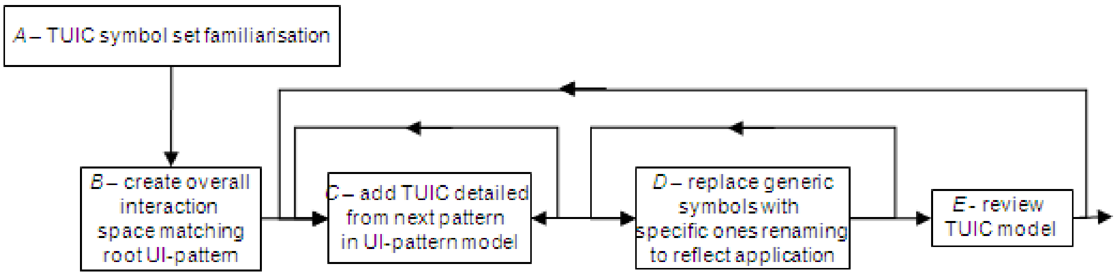


Figure A5.12 - Schematic of the pattern informed TUIC modelling method

Step A.	Become familiar with the TUIC symbol set.
Step B.	Starting with the top or root pattern of the UI-pattern model, examine the solution and the TUIC model diagram to get an overview of the TUIC model to be created. Draw the main interaction space and label it.
Step C.	Work down through the patterns using the diagram to guide the development of the TUIC model. Add CAP & navigation symbols to define details. In the initial stages use the generic components.
Step D.	As the modelling progresses, replace the generic components with specific ones where they make the intent of the design clearer. Rename interaction spaces and components to reflect the application's domain. Active material components should be used to replace equivalent material and tool components. Comments that clarify patterns used should be added.
Step E.	Review the model. This review may remove unnecessary layers of nested containers, group and reorder related symbols, and rename symbols. Comments can be added to make types of interactions and requirements clearer. Consider adding content details too.

Table A5.9 – Abstract prototyping method.

The steps for creating a TUIC model are shown in Figure A5.12 with each step expanded in Table A5.9. This method takes an essentially top-down approach to development. It is assumed that the design team developing the TUIC model is the same as the one that developed the UI-pattern model. If not the design team would need to become familiar with that model and the associated sub-set of patterns.

TUIC modelling starts with the designer becoming familiar with the TUIC symbol set. The subset used with the workbook example can be found in the appendices. Like the development of the UI-pattern model, over time familiarisation would not be necessary for all members of a design team.

The initial TUIC model

Every TUIC model has an enclosing container that is labelled to reflect the user interface being modelled. This defines the boundaries of the interaction space being modelled. In this case study the root pattern '02 High-density information' of the UI-pattern model representing the "Course Registration System". It becomes the enclosing context for the UI and completes step B.

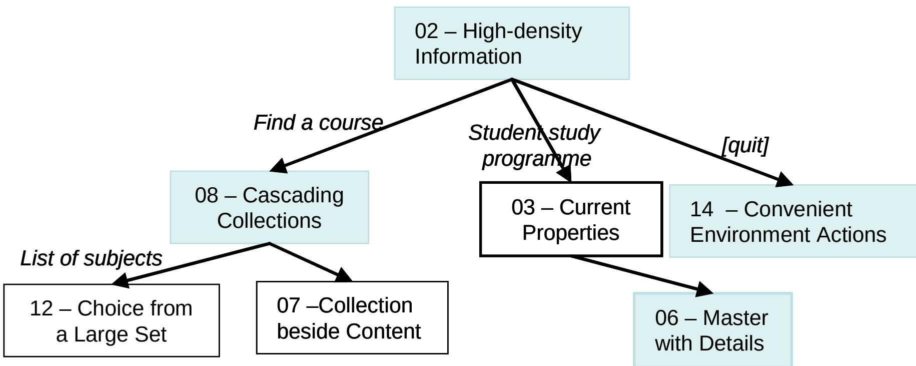


Figure A5.13 - Top levels of the UI-pattern model

Figure A5.13 shows the top levels of the UI-pattern model. Using the diagrams illustrating the next level of patterns as a guide the TUIC model's main structure can be created.

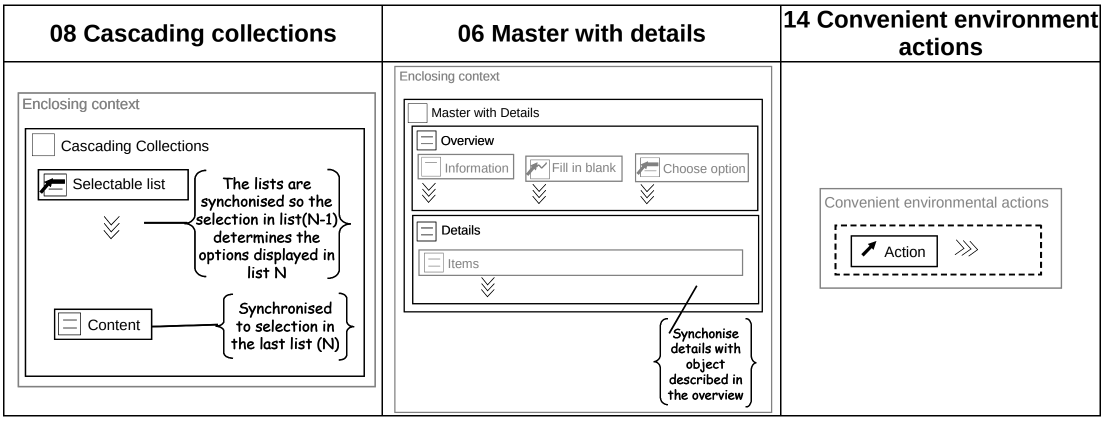


Figure A5.14 - Diagrams illustrating the essential aspects of each pattern's solution

The illustrating diagrams from these patterns are shown in Figure A5.14. The pattern '08 Cascading collections' represents the search space where a student should be able to find courses they want to enrol in. The pattern '03 Current properties' represents the interaction space for a student's enrolment records but as this pattern is instantiated by just '06 Master with details' it can be substituted simplifying the prototype. As there is only one action represented by '14 convenient environment actions' the enclosing grouping symbol can be removed. It represents the action to quit an enrolment session. Figure A5.15 shows the resulting structure for the TUIC model with some specific naming for interaction spaces and some general naming from the matching patterns.

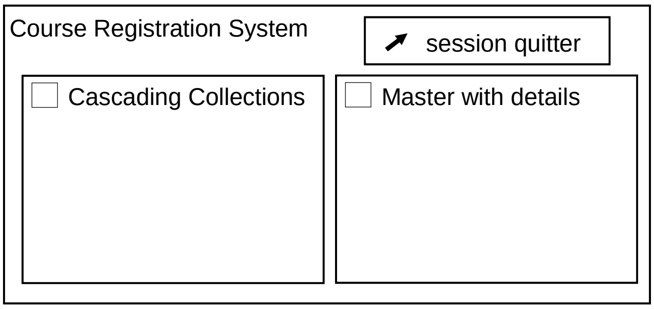


Figure A5.15 – An initial TUIC model using generic symbols and naming.

When comparing the diagrams in Figure A5.14 and Figure A5.15 notice that the internal structures of the diagrams illustrating the patterns shown in Figure A5.14 are not reproduced in Figure A5.15. The lower level patterns in the UI-pattern model will help guide the details for this specific TUIC model whereas those illustrating the patterns are generalisations.

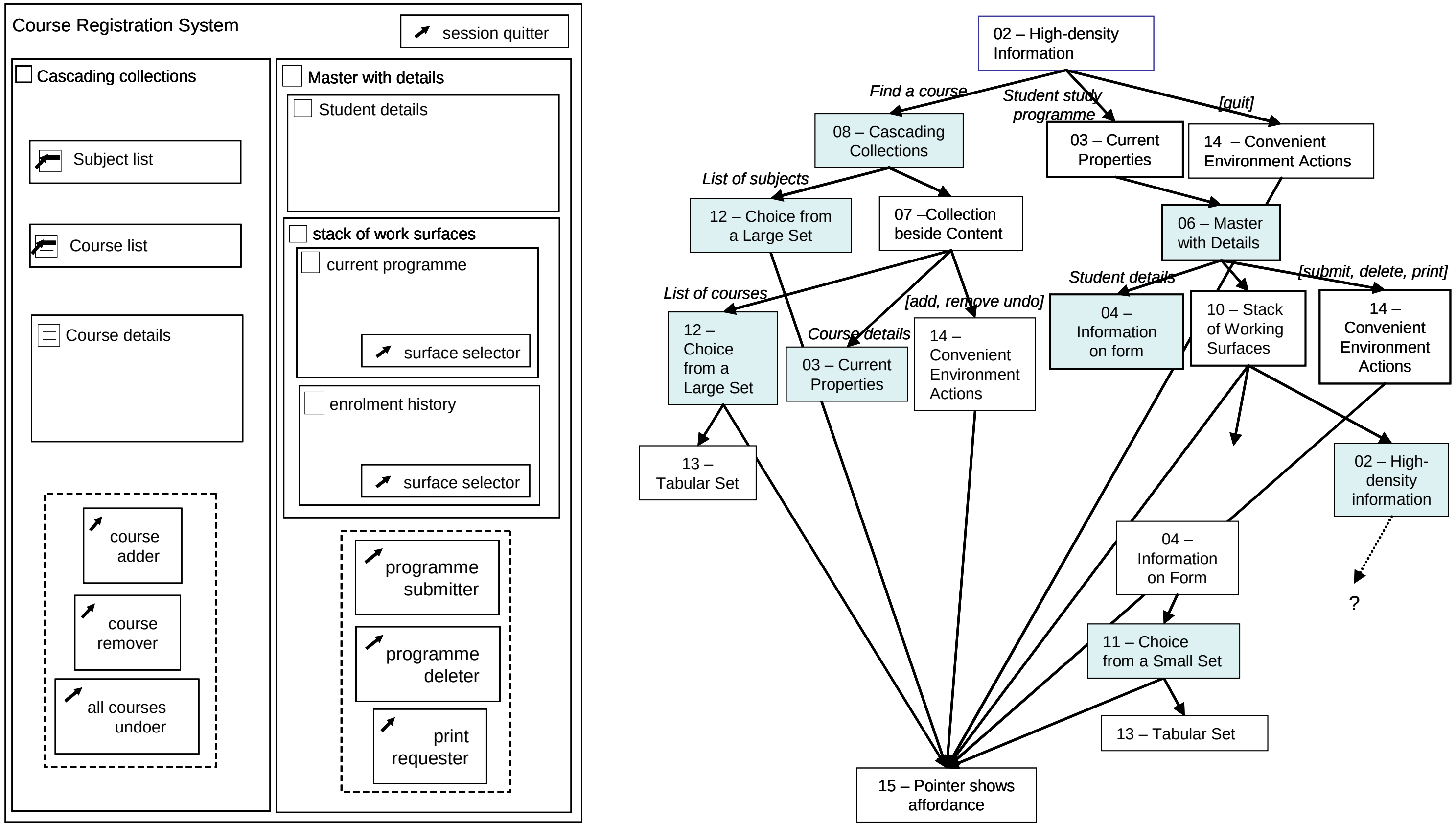


Figure A5.16 - Partial TUIC model with matching nodes shaded on the UI-pattern model.

Step C uses the next level of patterns to guide the addition of the two main container areas and one component as can be seen in Figure A5.15. The pattern '08 Cascading collections' is made up of two synchronised collections 'list of subjects' and 'list of courses' and the 'details of the selected course' shown by the naming in figure A5.15.

In Step D the designer replaces the generic names and symbols with ones that more accurately reflect the application domain. This step is often carried out in conjunction with step C. The method is iterative with these two steps repeatedly applied as the UI-pattern model is traversed. Figure A5.16 shows an almost complete TUIC model that represents the patterns of the coloured nodes in the adjacent UI-pattern model. This prototype still has many generic symbols and names. It also uses just generic CAP tools and materials.

Refining the TUIC model

Further refinements should replace some generic symbols with more specific ones as can be seen by comparing the TUIC models shown in Figure A5.15 and Figure A5.16. For example, 'Student details' CAP symbol[1] is refined to use the collection component. Some generic tools and materials can be replaced with active materials. For example, both the Subject list and the Course list have been changed to use the 'selectable collection' active material symbol in the refined TUIC model, Figure A5.17.

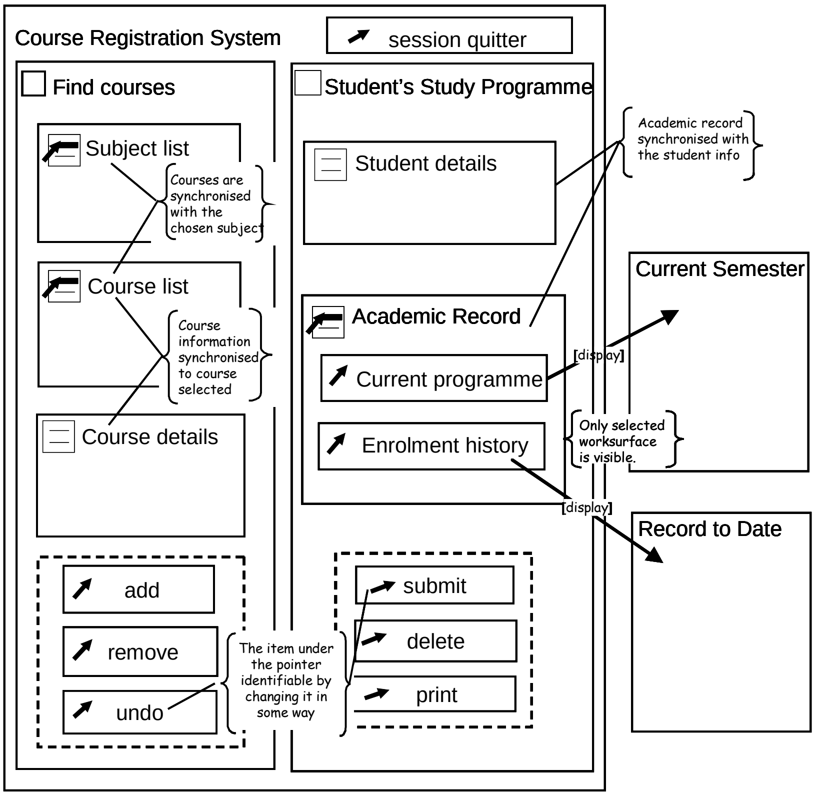


Figure A5.17 - Refined TUIC model.

Another refinement is to check the content of the patterns to identify alternative representations. For example, the space representing pattern '09 Stack of working surfaces' is not a good representation if additional field or attribute data is known and can be added to the model. Figure A5.17 uses an alternative TUIC model for this pattern. This representation makes the alternative interaction spaces much clearer and provides more space for adding further details. The naming has also been improved. Many of the TUIC model diagrams illustrating the patterns include comments. Appropriate comments associated with such patterns have also been added to Figure

[1] Reference: **Constantine and Lockwood, Ltd** http://www.foruse.com/articles

A5.17. For example, the comments that indicate which items are synchronised between subject and courses when modelling the pattern '08 Cascading collections'.

These steps are not applied in a linear fashion. The designer will move back and forward between steps C and D and the developing prototype will show a mix of generic and domain-specific names and symbols. Step E is a review process where more details can be added. The exercise to follow does not require this final review step nor is a lo-fidelity prototype developed but both are included in the appendices for completeness.

-------->> PLEASE STOP HERE TO DISCUSS ANY ISSUES OR IDEAS YOU MAY HAVE ABOUT THE METHOD AND THE UI-PATTERN MODEL <<-----

A5.6 TO DO

Please study the accompanying exercise example requirements and complete a partial UI-pattern model from the requirements.

A5.7 Acknowledgements

I wish to thank Associate Professor Chris Philips for making available the example case studies he used when teaching UI courses. His initial examples saved a considerable amount of work for which I am very grateful.

A5.8 Reference

Bayle, E., Bellamy, R., Casaday, G., Erickson, T., Fincher, S., Grinter, D., Gross, B., Lehder, D., Marmolin, H., Moore, B., Potts, C., Skousen, G., and Thomas, J., 1998. Putting it all together: Towards a pattern language for Interaction Design. SIGCHI Bulletin. 30(1), pp 17-24.

Constantine, L. 1998. Rapid Abstract Prototyping. Software Development. Available at <http://www.foruse.com/articles/canonical.htm> (October 1998)

Constantine, L., Windl, H., Noble, J., & Lockwood, L. 2003. From Abstraction to Realization: Canonical Abstract Prototypes for User Interface Design (No. 109): Constantine & Lockwood, Ltd.

Dearden, A., and Finlay, J. 2006. Pattern Languages in HCI: A Critical Review. Human-Computer Interaction, 21(1), pp. 49-102.

Erickson, 2000. Lingua francas for Design: Sacred Place and Pattern Languages. In: Proceedings of the conference on Designing Interactive Systems, Brooklyn, NY, pp 357-368.

Tidwell, J. last updated 2006. UI Patterns and Techniques. Retrieved December, 2006, from http://designinginterfaces.com/Visual_Framework

Tidwell, J. 2006. Designing interfaces. Sebastopol, CA: O'Reilly.

A5.9 Appendices

Reviewing the TUIC model

In step E the TUIC model is reviewed to check that all patterns have been modelled and that requirements have been met. Additional comments may need to be added to make some requirements clearer such as the comment indicating that a running total of the programme's cost should be displayed as the student builds it. The contents of the different interaction spaces, as shown in Table A5.3, can also be added

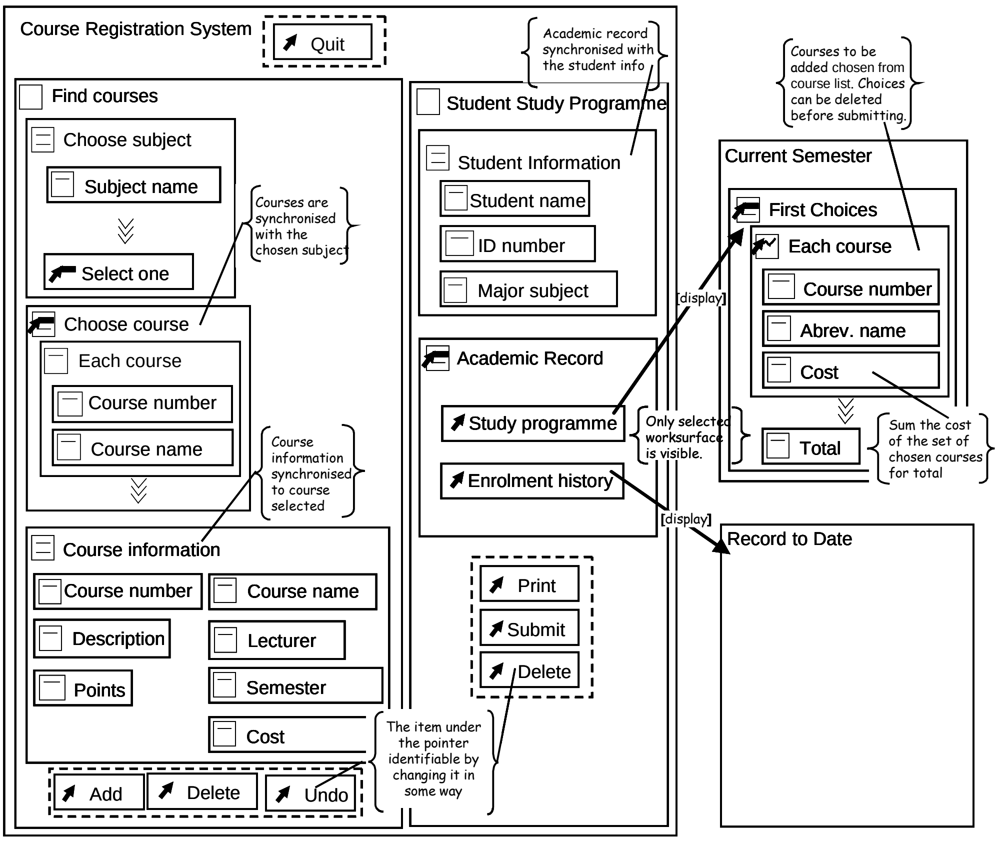


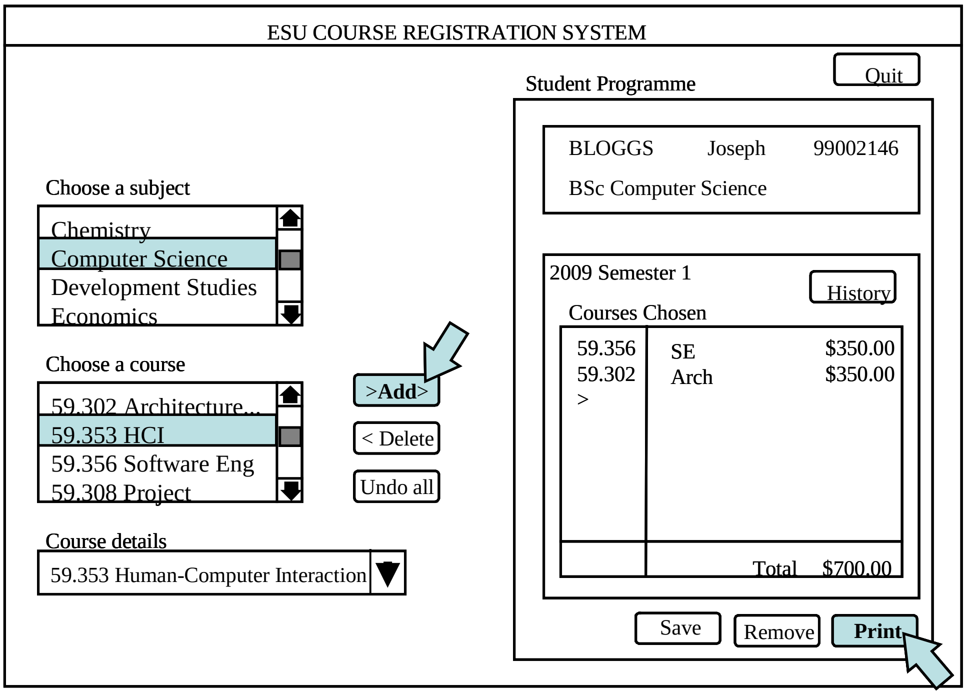
Figure A5.18 – TUIC model showing details of fields or attributes.

After Step E the almost completed TUIC model will be similar to that shown in Figure A5.18. It does not include details for the enrolment history because this is not covered by this example.

Visual Design

The following lo-fi visual design for this UI is conventional and has been provided for completeness. Design decisions have not been provided.

Example Lo-fi UI prototype



NOTE

The TUIC symbol set and two papers were also included for the Case study participants.

Appendix A6: Studies One, Two & Three - Exercises

A6.1 Studies 1 & 2 - Instructions

The terminology used in the UI-pattern modelling method is that used in Study Two. The original wording can be found in Chapter Five. Otherwise the instructions were the same for all UI-pattern modelling exercises.

EXERCISE INSTRUCTIONS – UI-pattern modelling

In this exercise you are asked to build a user interface pattern language model (UI-Pattern model) to describe the user interface illustrated. A pattern-language-model is a set of linked UI patterns that build a network of patterns that describes the UI interface. A pattern-language-model can be described as a set of one or more overlapping hierarchies.

Artefacts

They are:

⇒ An illustration of a user interface.
⇒ Access to a set of UI Patterns.
The patterns are numbered so that generally the earlier patterns refer to the overall interface and the last to more detailed features of an interface. The patterns may refer to other patterns that are not in the subset you are using for this exercise.
⇒ An exercise sheet with attached coversheet.
⇒ A questionnaire attached behind the exercise sheet.

Method

1. Write your name and ID on the cover pages for the exercise artefacts.

2. Use the following method to create a UI-pattern model

Step A.	Become familiar with the subset of UI patterns.
Step B.	Examine the illustration of the user interface for the exercise.
Step C.	Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information the user interface (UI) deals with.
Step D.	From the list of patterns that this pattern references, select those that best describe some other feature of the user interface you have been asked to model.
Step E.	Continue identifying patterns that best describe a feature of the user interface from each selected pattern’s reference section until you can’t select any more
Step F.	Using your selected patterns create a diagram showing those patterns and connected with lines to indicate how they are linked into a UI-pattern model. Use the context and reference sections of your patterns to check for missing links or patterns.

3. When you have finished creating the pattern-model complete the associated questionnaire.

Notes

You are encouraged to consult with your partner as you develop the UI-pattern model but each of you should hand in your own version of this model.

Please hand in all the patterns and the UI illustration with your exercise sheet and questionnaire.

EXERCISE INSTRUCTIONS – Conceptual UI modelling

In this exercise you are asked to build a conceptual UI model of a user interface based on a UI-pattern model that describes the given user interface illustration. It is called a **T**eaching **UI C**onceptual (TUIC) model and is created using canonical abstract prototype (CAP) and navigation components.

Artefacts

They are:

⇒ The set of nineteen UI Patterns from the first exercise.
The solution of each pattern is illustrated using a canonical abstract prototype.
⇒ An identification sheet of TUIC symbol set used in diagrams illustrating the solutions in the patterns.
⇒ A list of the patterns summarising context and reference lists.
⇒ An annotated illustration of the example user interface.
⇒ An exemplar UI-pattern model.
⇒ An exercise sheet with attached coversheet.
⇒ The TUIC component questions attached behind the exercise sheet.

Method

1. Write your name and ID on the cover page for the exercise artefacts.

Step A.	Find the subset of UI patterns used in the UI-pattern model and become familiar with how they describe the example UI.
Step B.	Become familiar with the TUIC symbol set.
Step C.	Starting with the top pattern, examine the solution and the TUIC model diagram to get an overview of the model you are to create. Draw an outline space and label it.
Step D.	Work down the patterns using the TUIC model diagram to guide your modelling. Use generic symbols for higher level patterns.
Step E.	Use lower level patterns to add details and navigation components. Refine generic CAP symbols with specific CAP symbols.
Step F.	Name all components on your TUIC model to reflect the example UI.

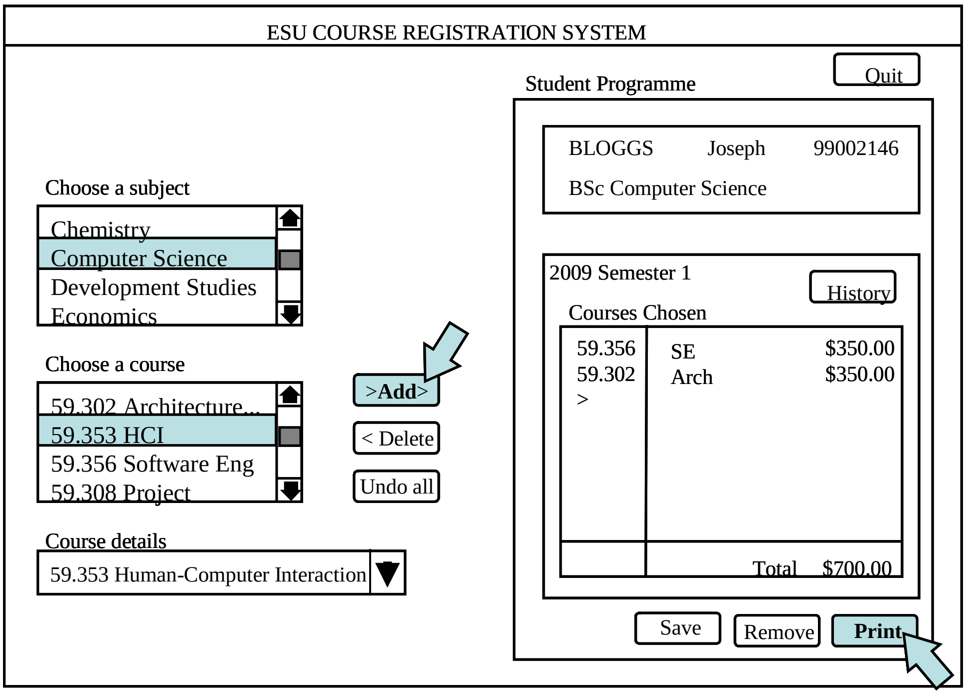
2. When you have finished creating the TUIC model please complete the associated sets of questions.

Notes

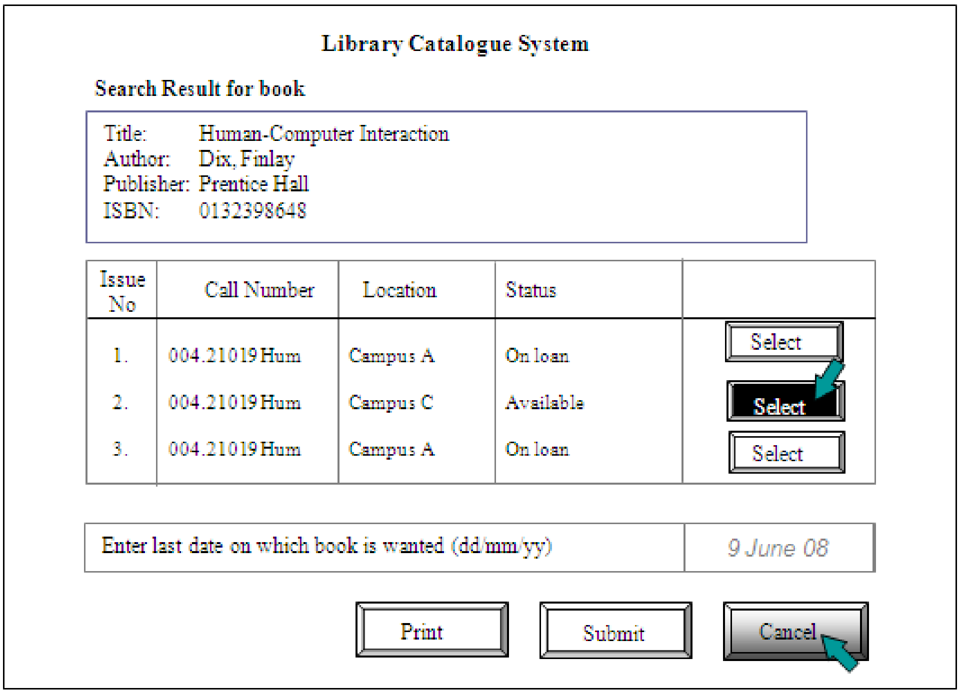
You are encouraged to consult with your partner as you develop the TUIC model but each of you should hand in your own version of this model and complete the questions independently.

Please hand in all the patterns and the UI illustration with your exercise sheet and questions.

A6.2 Study 1 – Exercise Two Example UI

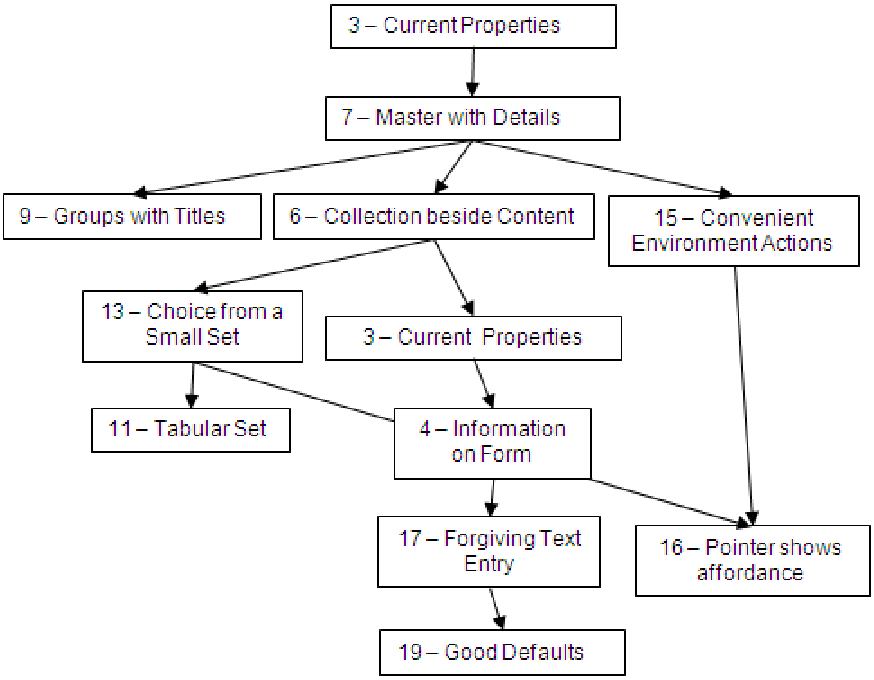


A6.3 Studies 1 & 2 – Exercise One Example UI



A6.4 Study 2 – Exercise Two Example UI-pattern model

This UI-pattern model represents the Library Catalogue System UI above.



A6.5 Study 3 – Example

Conceptual Design using UI-pattern Modelling and TUIC Modelling

The goal for this exercise is to create two conceptual user interface models for the system outlined in the following requirements documentation. When applying each method please remember that the intent is that it be applied by a design team that includes potential users of the user interface or that it is being applied by a team of UI design students.

INITIAL REQUIREMENTS DOCUMENTATION

The following initial requirements documentation is incomplete. Sufficient details have been provided so that a UI-pattern model and TUIC model of the reservation stage of the room registration system can be created.

A6.6 Narrative Overview – Hotel Room Registration System

In order to streamline the registration of guests, the Kiwi Hotel plans to install an on-line reservations system. The first stage is a room reservations system for use by hotel reception staff. These staff members are reasonably computer literate.

A6.7 Requirements

Functional Requirements

- The identity of the staff member responsible for each stage of a transaction must be recorded as should the time and date when the stage is confirmed.
- The criteria for the list of rooms should include: room number, capacity code, availability and cost. Availability is defined as: status, start date, end date, number of nights; where status indicates whether the room is: available, booked, occupied, needs a special clean or being redecorated. The capacity code should indicate the number and types of bed.
- Provide the staff member with a facility to reduce the list of potential rooms based on additional criteria such as: required number and type of beds required, type of access, view or number of adjacent rooms.

- The staff member should be able to display additional information about a room. This information should include English descriptions explaining any codes used as search criteria. Information to include: number and types of bed, access type (wheel chair, no stairs, children, special bathroom etc.), view code including a description and a list of facilities such as fridge, kettle sofa etc.
- Allow staff members to inspect the availability history of any room as necessary.
- Ensure the staff member saves all details at each stage of the transaction.
- Staff members should be able to retrieve all details associated with a client at each phase of the room registration transaction (reservation, check-in, check-out and payment completed).
- Staff member should be able to print each phase of a room registration transaction.
- …

Non-Functional Requirements

Staff members use this system regularly throughout their shift at the reception desk. The manager will also be called in periods when activity is very busy to use the system probably about once or twice a week. As reception staff are also dealing with other requests from clients, often at the same time, it is important that the system is easy and intuitive to use. Trainee staff will also be expected to use the system when activity is reasonably quiet.

The system should:

- Present information in a clear, well organised and concise manner.
- Provide good support and feedback at each stage.
- Support an efficient and flexible dialogue, including a good search capability for rooms.
- Be satisfying to use. The receptionist should feel confident they have successfully met the needs of the hotel's clients.
- …

A6.8 Scenarios

The following scenario is typical of the use of the system by a staff member who has been employed at the hotel for some time.

Receptionist

John, one of the hotel receptionists, wishes to make a reservation for the Smiths, a family who have made contact by phone. They are already travelling, and are seeking accommodation for one night later that day. He first searches through the list of rooms', limiting the search to rooms available on that day, and of adequate size. The system displays details of each available room. John relays the information to the Smiths, who place a further constraint relating to cost. Three suitable rooms are available, and the one with the best view is selected. John requests credit card details and a contact phone number. He then creates a reservation for the Smiths for the selected room.

Later in the day the Smiths check in. John updates the reservation, including more complete details of the guests.

The following morning the Smiths arrive at the reception desk to check out. John checks whether additional items have been charged to their room. He updates the accommodation transaction and creates a final account and prints the full account. He checks with the Smiths that it is correct. Then he records that they accepted the account details, bills their credit card, and prints a receipt.

Other scenarios …

A6.9 Structured User Role Model

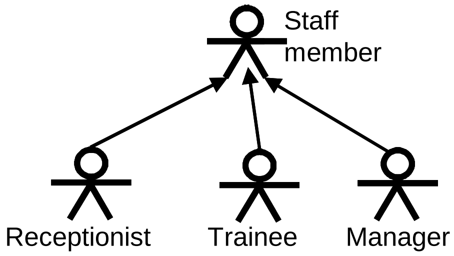


Figure A6.1 - Role map showing the different types of staff member.

Receptionist

Goal:	**to respond-to and register each stage of a clients' accommodation cycle**
Background:	completed hotel's in-service receptionists training ten years on the job experience computer literate
Domain Knowledge:	Familiar with the hotel's hotel room registration procedures Understands the interactions between the checking-in and checking-out of clients and other hotel operations such as house-keeping and catering Experienced with meeting unusual customer requirements. Called on by less experienced staff when a client's requirements are non-standard
System:	Has experienced three upgrades to current system
Proficiency:	intermediate to experienced
Interaction:	intensive use each workday (uses the system frequently during each shift) user driven
Information:	flows both ways information out can be comparatively complex information in is not excessive in volume
Functions:	no special functions
Usability:	good quality of presentation of information good support and feedback efficient, flexible and reliable satisfying

Table A6.1 - Details of role 'Receptionist' which specialises Staff member

Other roles …

A6.10 Task Model

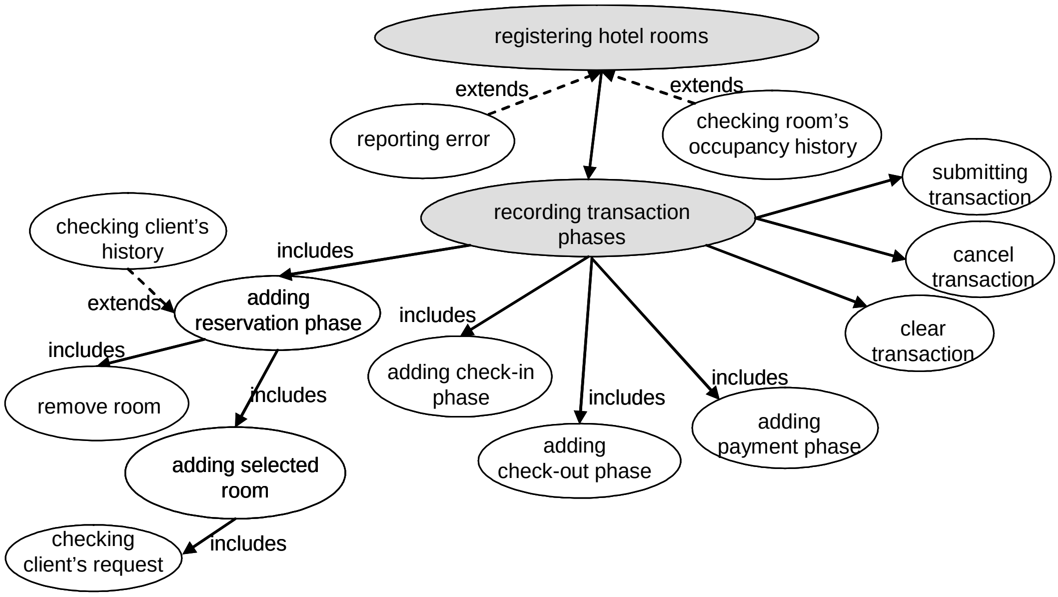


Figure A6.2 - Use case map with those shaded to be modelled

A6.11 Use case details

Once a use case is completed it is examined to identify actions and interaction spaces. These are itemised in the accompanying table.

Registering hotel rooms

Registering a hotel room includes the check-in and check-out procedures as well as the reservation procedures **ROLES SUPPORTED:** **receptionist, trainee, manager** **PRE-CONDITIONS:** **staff member is logged in, system initialised**	
USER INTENTIONS	**SYSTEM RESPONSIBILITIES**
optionally at any point do: checking room's occupancy history	**optionally at any point do:** reporting error
	1 present staff member's name & id 2 present today's date and time 3 present adding reservation interaction space 4 offer room list pre-ordered to show those currently available for occupancy from today's date in find room space
5 **optionally**: [sorting list of available rooms based on a basic room field]	
	6 change order of available rooms list based on sort criteria 7 present re-sorted room list.
8 **optionally**: [advanced-sorting room list based on non-basic field]	
	9 activate selected sort field with room list. 10 **if necessary:** request sort order for activated field 11 change order of available rooms list based on sort criteria 12 offer resorted room list.
13 **optionally**: [starting new sort]	
	14 **do**: save sort onto sort history list 15 offer room list pre-order to show those currently available for occupancy
16 **optionally**: [retrieve saved sort]	
	17 **either**: previous saved sort is presented **or**: next saved sort is presented **or**: notify user no more saved sorts
18 **optionally**: [selecting a room]	
	19 present selected room's non-basic details
20 **optionally do**: recording transaction phases;	
21 [quitting the session]	
	22 **either**: all suspended reservation have been submitted and end session processes **or**: notify user and provide option to submit
POST-CONDITIONS: staff member is logged out	

Actions	**Interaction spaces**
	Staff details with date & time
	Find rooms
Select room	» List of rooms
Check room’s occupancy history	» Selected room’s details
	» Selected room’s occupancy history
Reorder room list	
Advanced reorder room list	
New sort	
Back through saved sorts	
Forward through saved sorts	Transaction space
	Error reports
	Request sort order
	Notify user of need to submit or suspend transaction data
Quit registering hotel rooms	Notify that transactions must be submitted or deleted

Table A6.2 - Inventory of actions & spaces for use case “registering hotel rooms”.

Recording transaction phases

To register the different phases of the registration of rooms to clients ensuring each stage is submitted successfully **ROLES SUPPORTED:** **receptionist, trainee, manager** **PRE-CONDITIONS:** **previous activity has been submitted or cancelled** **current sort saved and room list re-ordered as for new sort**	
USER INTENTIONS	**SYSTEM RESPONSIBILITIES**
	optionally at any point do: reporting error
	1 staff member’s id added to transaction 2 today’s date and time added to transaction 3 present reservation phase in transaction space
4 **optionally do:** <u>adding reservation phase</u>; <u>adding check-in phase;</u> <u>adding check-out phase;</u> <u>adding payment phase</u>	
5 **optionally**: [specifying that transaction fields be printed]	
	6 format the transaction fields for printer and send, notifying users of progress
7 **optionally**: [specifying that current transaction phase be submitted]	
	8 **either**: not all required information has been supplied so notify user **or**: update phase to clients registration transaction record recording transaction phase
9 **optionally do:** [specifying that current transaction phase be cancelled]	
	10 **either** client has not checked in **do**: <u>cancel reservation</u> **or**: notify client already occupying room
POST-CONDITIONS: transaction details submitted or cancelled and all fields cleared	

Actions	Interaction spaces
Adding reservation phase Adding check-in phase Adding check-out phase Adding payment phase Print transaction phase Submit transaction phase Cancel transaction	Transaction phase space Error reports Notify information missing Notify submission progress and success Notify client already checked in

Table A6.3 - Inventory of actions & spaces for use case "recording transaction phases"

Other use cases …

A6.12 Details of Interaction Spaces

Spaces	Content
Staff details	Staff name Staff id Date Time
List of rooms	Room number Capacity code Availability status Date in Date out Room rate
Room details	Facilities list Access type Access description View code View description
Room's occupancy history	Availability status Date in Number of nights Family-name of occupier Comments
Error reports	As appropriate
Notification dialogues	…

Table A6.4 - Interaction space details for use case "registering hotel rooms.

Spaces	Content
Transaction phase space	phase name

Figure A6.3 - Interaction space details for use case "recording transaction "

Other use case inventories

Reservation
Check-in
Check-out
Payment
…

A6.13 TO DO – UI-pattern Model

The exercise is to develop a UI pattern model for the hotel registration system defined above, using the methods as demonstrated in the introductory example. Only the core aspects of the room reservation phase making up a room registration transaction is to be modelled.

The UI-pattern Modelling

Use the following method to create a UI-pattern model for the system detailing the registration phase but not the check-in or check-out phases.

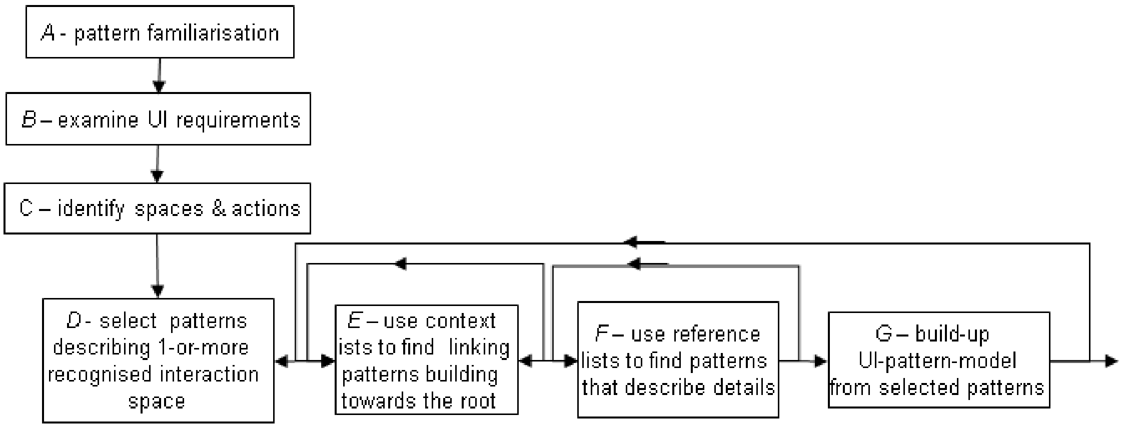


Figure A6.4- Schematic of the UI-pattern model building method

Step A.	Become familiar with the UI patterns.
Step B.	Examine the UI requirements documentation.
Step C.	Identify actions and interaction spaces then group related interaction spaces and any associated actions.
Step D.	Select patterns that seem to best describe one or more of the interactions spaces.
Step E.	Check each pattern's Context section and select a pattern from the list that gives the best overall description. This description may include patterns already selected. A pattern may be mentioned in the context section of one or more of the selected patterns. Use the context section to find common patterns until one of these higher level patterns can be identified as the root (1, 2 or 3) of the UI-pattern model
Step F.	Check each pattern's Reference section to identify patterns that best describe the details and additional features required for the user interface.
Step G.	Using selected patterns, create a diagram consisting of patterns connected with lines to indicate how they are linked into a structure. Use the context and reference sections of patterns to check for missing links or patterns.

Table A6.5 - Method for creating a UI-pattern model.

--------→> PLEASE STOP HERE TO COMPLETE QUESTIONAIRE AND DISCUSS EXPERIENCE <←-----

A6.14 TO DO – TUIC Model

The exercise is to develop a TUIC model for the hotel registration system using the UI-pattern model as a guide. Please use the methods demonstrated in the introduction. There is only time to complete the first four steps of this method so details for individual fields in the interaction spaces will not be added.

The TUIC Modelling

Use the following method to create a TUIC model but there will probably not be time to complete step E so do not add additional comments or content details.

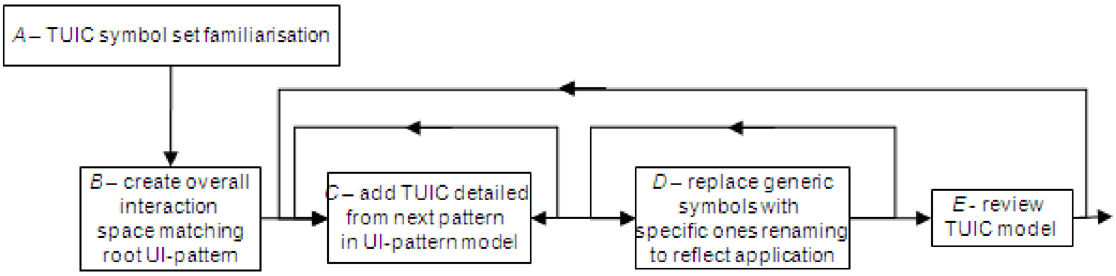


Figure A6.5 - Schematic of the pattern informed abstract prototyping method

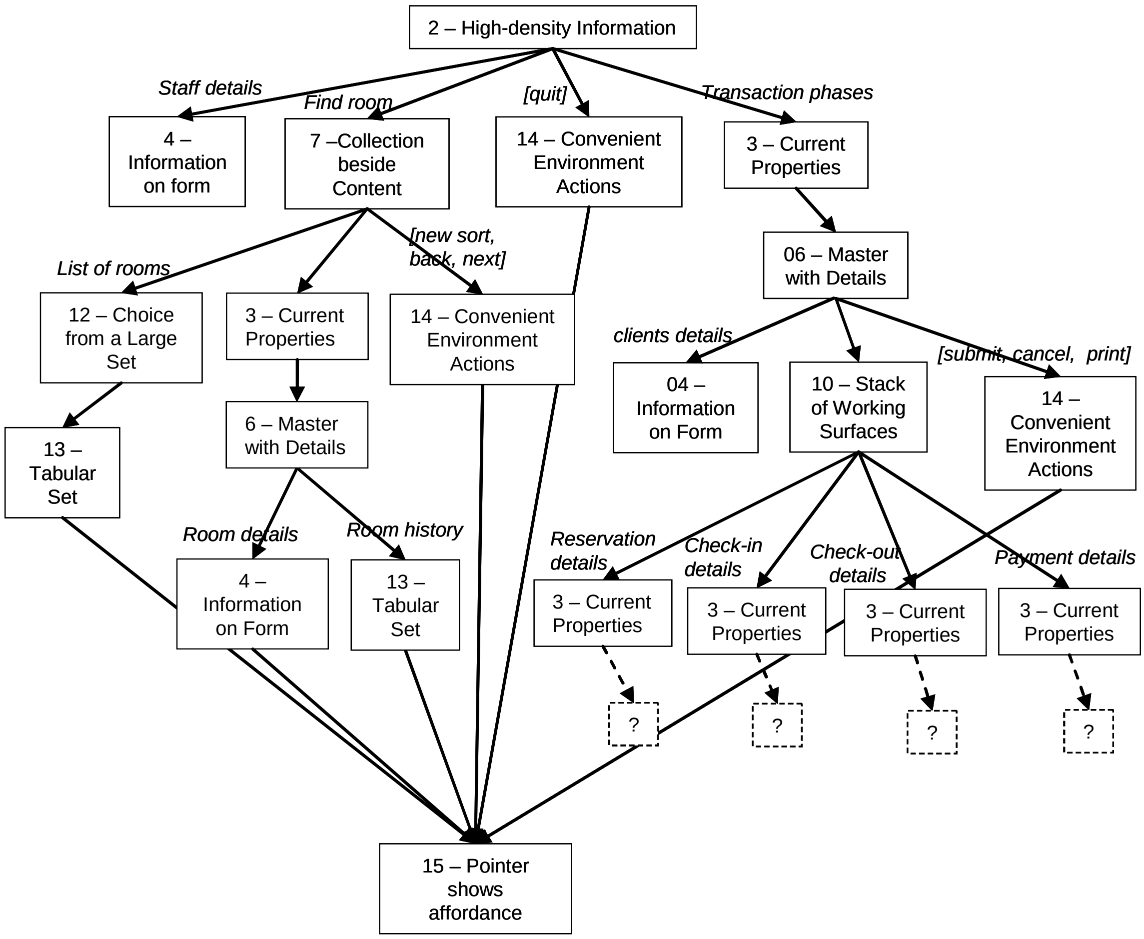
Step A.	Become familiar with the TUIC symbol set.
Step B.	Starting with the top or root pattern of the UI-pattern model, examine the solution and the TUIC model diagram to get an overview of the TUIC model to be created. Draw the main interaction space and label it.
Step C.	Work down through the patterns using the diagram to guide the development of the TUIC model. Add CAP & navigation symbols to define details. In the initial stages use the generic components.
Step D.	As the modelling progresses, replace the generic components with specific ones where they make the intent of the design clearer. Rename interaction spaces and components to reflect the application's domain. Active material components should be used to replace equivalent material and tool components. Comments that clarify patterns used should be added.
Step E.	Review the model. This review may remove unnecessary layers of nested containers, group and reorder related symbols, and rename symbols. Comments can be added to make types of interactions and requirements clearer. Consider adding content details too.

Table A6.6 – Abstract prototyping method.

--------→> PLEASE STOP HERE TO COMPLETE TWO QUESTIONAIRES AND DISCUSS EXPERIENCE <←-----

A6.15 Example UI-pattern Model for the Hotel Reservation system

The case study participants were provided with the exemplar UI-pattern model below to work from for the third session rather than use the one they had created during the second session.



Appendix A7: Studies One & Two – Exercise Worksheets

A7.1 Study One & Two - Cover sheet for UI-pattern Modelling

Note that the objectives were re-worded on the coversheet so students were not alerted to the importance of the illustrations.

IDENTIFYING COVER SHEET

This study has the following objectives:

- Determine whether UI patterns are an acceptable medium for presenting UI information to students.
- Investigate suitable formats for presenting the information within UI patterns and pattern languages.
- Determine whether the proposed method can successfully guide students in creating a UI-pattern model.
- Discover whether building a UI-pattern model develops student understanding of UI patterns and pattern language structure.

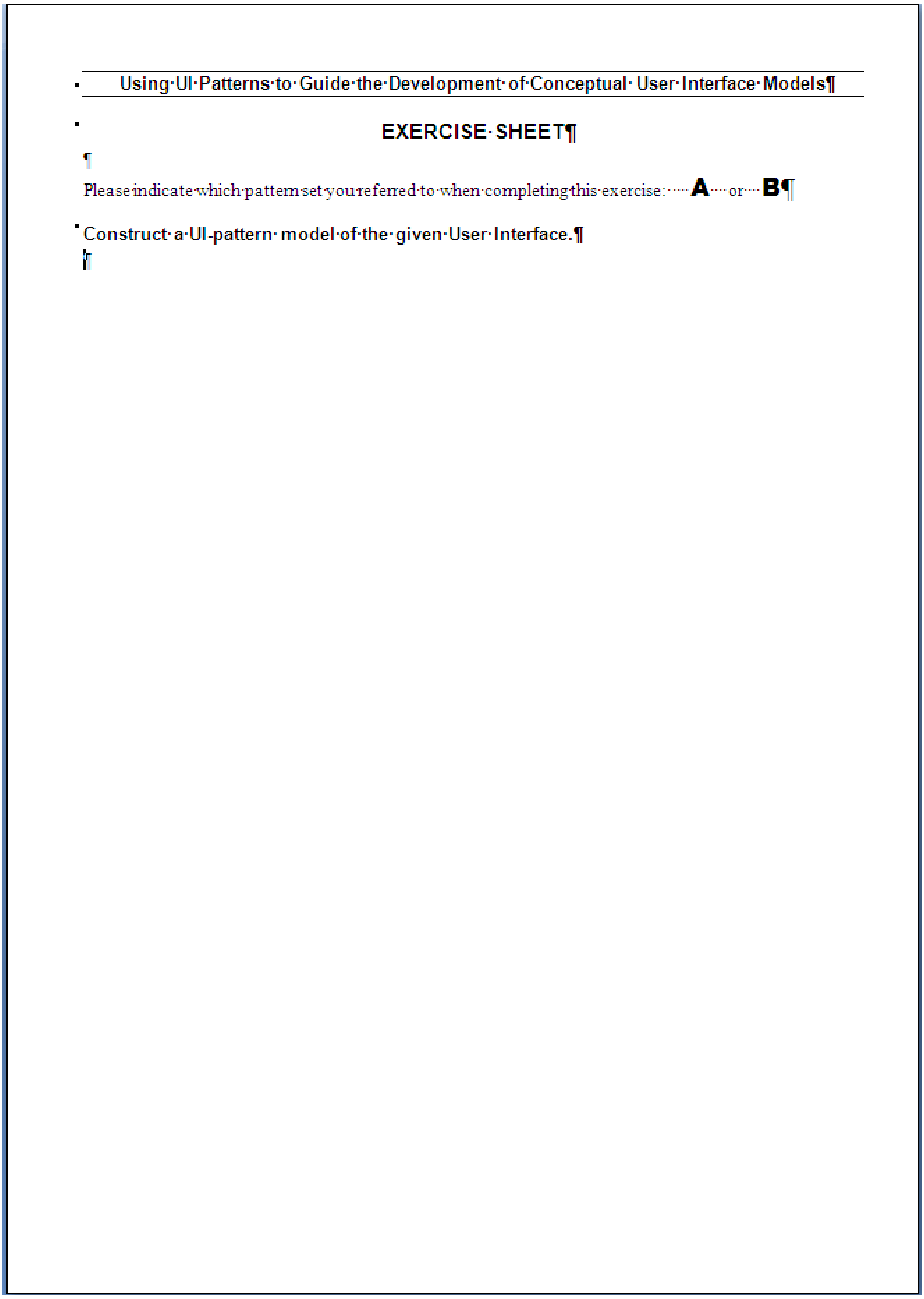
Once the different artifacts created during the experiment have been collated with the material identified by this page, this page will be removed and destroyed so that the research data becomes anonymous.

Name: ....

ID: ....

The results from the analysis of the experiment will be available from the CS office in AH3.91 on request. Participants will have to present their ID card to prove that they were enrolled in this class and therefore have a right to the information.

A7.2 Study One - Worksheet for UI-pattern Models



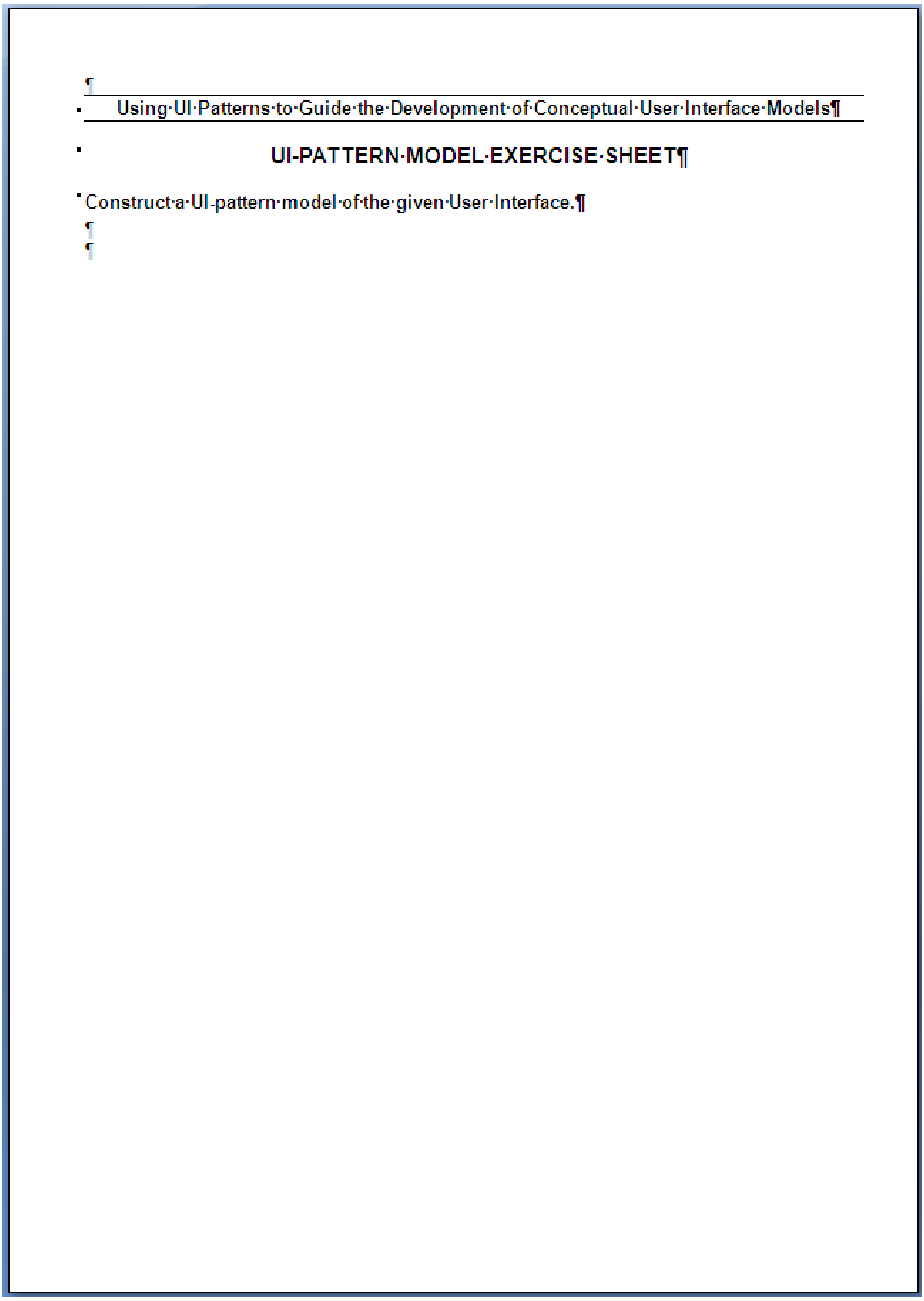
Using UI Patterns to Guide the Development of Conceptual User Interface Models

EXERCISE SHEET

Please indicate which pattern set you referred to when completing this exercise: **A** or **B**

Construct a UI-pattern model of the given User Interface.

A7.3 Study Two - Worksheet for UI-pattern Models



Using UI Patterns to Guide the Development of Conceptual User Interface Models

UI-PATTERN MODEL EXERCISE SHEET

Construct a UI-pattern model of the given User Interface.

A7.4 Study Two – Cover sheet for UI-pattern Models

IDENTIFYING COVER SHEET

This study has the following objectives:

- Determine whether the proposed method will support students using UI patterns as a guide to the creation of TUIC models.
- Determine whether UI patterns can be used by student UI designers to gain new UI knowledge.

Once the different artifacts created during the experiment have been collated with the material identified by this page, this page will be removed and destroyed so that the research data becomes anonymous.

Name: ....

ID: ....

The results from the analysis of the experiment will be available from the CS office in AH3.91 on request. Participants will have to present their ID card to prove that they were enrolled in this class and therefore have a right to the information.

A7.5 Study Two - Worksheet for TUIC Model

Using UI Patterns to Guide the Development of Conceptual User Interface Models

UI-CONCEPTUAL MODEL EXERCISE SHEET

Construct a UI-conceptual model of the given User Interface based on the exemplar UI-pattern model.

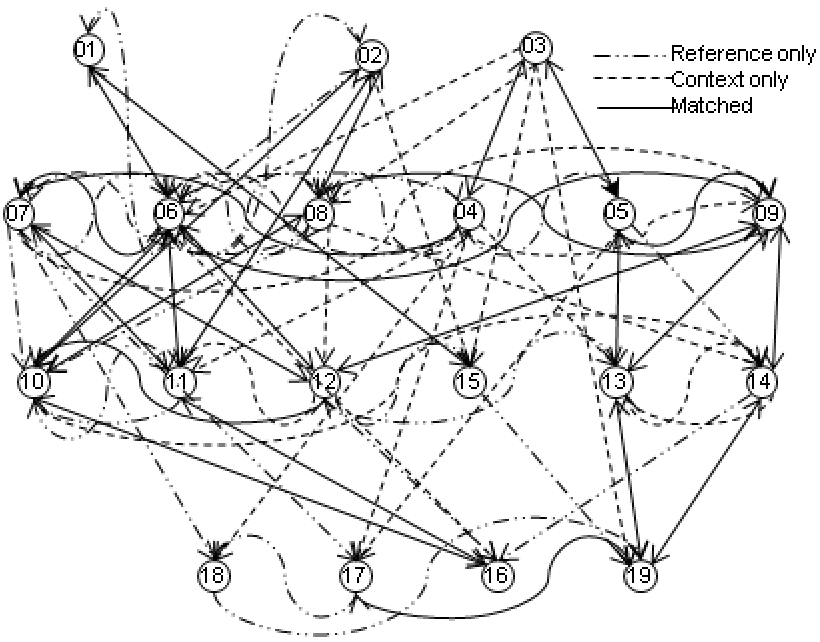
Appendix A8: TUI – Teaching UI Pattern Language

The following nineteen patterns are images of those used for Study Three.

A8.1 Pattern Maps

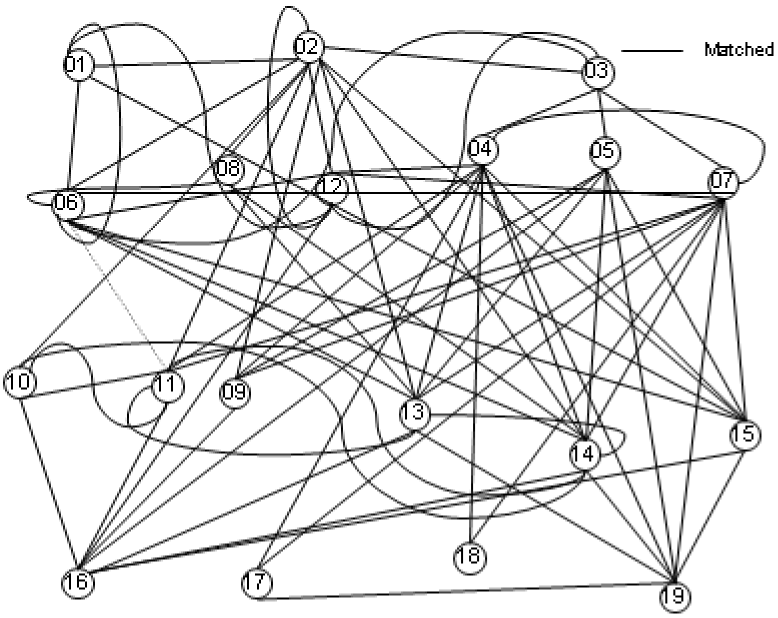
In the experiments the structure of the pattern language changed.

Study One



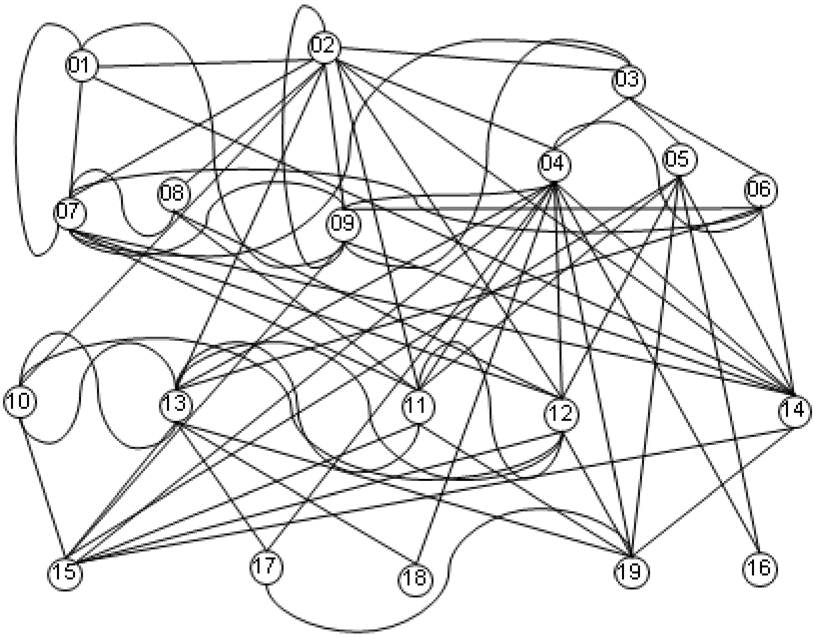
01 – Narrative	11 - Tabular Set
02 - High-density Information	12 - Stack of Working Surfaces
03 - State Information	13 - Choice from a Small Set
04 - Transaction by Form	14 - Choice from a Large Set
05 - Control Panel	15 - Convenient Environment Actions
06 - Collection beside Content	16 - Pointer Shows Affordance
07 - Master with Details	17 - Forgiving Text Entry
08 - Cascading Collections	18 - Input Prompt
09 - Groups with Titles	19 - Good Defaults
10 - Hierarchical Set	

Study Two



01 – Narrative 02 - High-density Information 03 – Current Properties 04 - Information on Form 05 - Control Panel 06 - Collection beside Content 07 - Master with Details 08 - Cascading Collections 09 - Groups with Titles 10 - Hierarchical Set	11 - Tabular Set 12 - Stack of Working Surfaces 13 - Choice from a Small Set 14 - Choice from a Large Set 15 - Convenient Environment Actions 16 - Pointer Shows Affordance 17 - Forgiving Text Entry 18 – Ask for Input 19 - Good Defaults

Study Three



01 - Narrative	11 - Choice from a Small Set
02 - High-density Information	12 - Choice from a Large Set
03 – Current Properties	13 - Tabular Set
04 - Information on Form	14 - Convenient Environment Actions
05 - Control Panel	15 - Pointer Shows Affordance
06 - Master with Details	16 - Groups with Titles
07 - Collection beside Content	17 - Forgiving Text Entry
08 - Cascading Collections	18 – Ask for Input
09 - Stack of Working Surfaces	19 – Good Defaults
10 - Hierarchical Set	

A8.2 The Narrative Pattern Set

To create a narrative pattern set remove both the diagrams illustrating the solution and the images for each example. The caption for each example should have the prefix reference to the figure and number removed so the examples become just a list.

The first example in the list should be moved to the beginning of the pattern before the Context section and be given a heading “Illustration.”

A8.3 The Illustrated Pattern Set

To create an illustrated pattern set remove the TUIC model diagram illustrating the solution. Move the first image in the examples list to the beginning of the pattern and give it a heading “Illustration”. The size of the image can be increased but not so large that the pattern takes up more space than two sides of an A4 sheet.

A8.4 The Diagrammed Pattern Set

The diagrammed pattern set follows.

The Patterns List

	Context links	Reference links
01 - Narrative	2, 7, 9	7, 14
02 - High-density Information	9	1, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14
03 – Current Properties	2, 7, 9	4, 5, 6
04 - Information on Form	2, 3, 6	9, 11, 12, 13, 14, 15, 16, 17, 18, 19
05 - Control Panel	3	11, 12, 14, 15, 16, 19
06 - Master with Details	3	4, 7, 9, 13, 14
07 - Collection beside Content	1, 2, 6, 8	1, 3, 9, 11, 12, 14
08 - Cascading Collections	2	7, 11, 12
09 - Stack of Working Surfaces	2, 4, 6, 7	1, 2, 3, 14, 15
10 - Hierarchical Set	2, 12, 13	13, 15
11 - Choice from a Small Set	2, 4, 5, 7, 8	13, 15, 19
12 - Choice from a Large Set	2, 4, 5, 7, 8	10, 11, 13, 15, 19
13 - Tabular Set	2, 4, 6, 10, 11, 12	10, 15, 17, 18, 19
14 - Convenient Environment Actions	1, 2, 4, 5, 6, 7, 9	15, 19
15 - Pointer Shows Affordance	4, 5, 9, 10, 11, 12, 13, 14	
16 - Groups with Titles	4, 5	
17 - Forgiving Text Entry	4, 13	19
18 – Ask for Input	4, 13	
19 – Good Defaults	4, 5, 11, 12, 13, 14, 17	

NOTE: The following pages show screen images of the relevant parts of the pattern pages because the margins used for the original patterns will not fit within those required for this thesis.

01 - Narrative

Context

There is a need to convey information to the user; the information is closely interrelated, but maybe of diverse kinds, and there may be some subjectivity involved.
Like most patterns that can provide an overview description of a user interface this pattern can also be used for the content part of the pattern COLLECTION BESIDE CONTENT or for information on a surface in a STACK OF WORKING SURFACES and also as part of a display of HIGH-DENSITY INFORMATION.

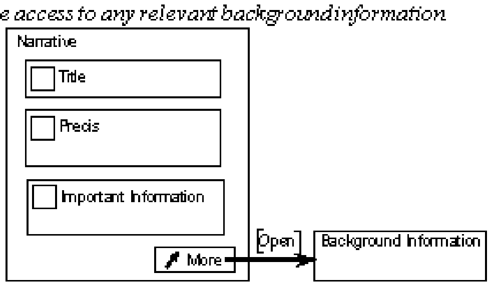
Problem

In what form should longish chunks of narrative information be presented to the user?

Solution

This approach to presenting information is basically relatively simple:

1. *Create a clear title*
2. *Provide a précis identifying the most important issues first;*
3. *Follow up with the most important supporting information;*
4. *Finally provide access to any relevant background information*



Reference

It is important that the reader know how to access the information so ensure CLEAR ENTRY POINTS. Use EXTRAS ON DEMAND to access background information. Much narrative has structure so a document map can be helpful using Collection beside content. It is important that actions such as printing that a user is likely to want to carry out are clearly accessible Convenient Environment Actions.

Forces

1. Many kinds of information are easier to absorb and remember when they are conveyed via natural language than when they are presented as raw data.
2. Many people find narrative to be more pleasant than raw data or symbolic representations.
3. People moving around the web skim for information or key words. If the main theme of the information is not quick and easy to grasp, it is usually not read.
4. Natural language presented as "straight text" has to be read or skimmed linearly to find specific information.
5. Natural language can convey subjectivity -- values, interpretations, and opinions -- in a way that no other information presentation can.
6. The artefact's or author's point of view should sometimes be explicitly acknowledged, such as when credibility is an issue; narrative can do this well, whereas other forms of data presentation are more anonymous.

Discussion

Use all you learned in high-school English class about good writing. Users will most likely be skimming the text for information or key words. Use such devices as colour, fonts, and white space to set off items of interest. Then look at the techniques used by journalist when writing. Jacob Neilson for example recommends using the Inverse pyramid writing style http://www.useit.com/alertbox/9606.html

Other techniques that should be incorporated are to first create a concise but descriptive title for the information According to van Duyne et al. it should be no more than ten words. Also the writing should be in short sentences using a relatively straight forward vocabulary. When providing opinion avoid using hype and ensure that you are providing rational comment. For the middle section of important points supporting the conclusions consider using bulleted lists that may have links to detailed support information on following pages.

Brenda Laurel's quote on the use of narrative in an interactive artifact

"Narrative includes both the story being told (content) and the conditions of its telling (structure and context). ... Within that [narrative] framework, interface designers can adopt strategies from narrative theory, such as including multiple representations of events and information, or using characters as a means of representing material with an explicitly acknowledged point of view." (pg 182)

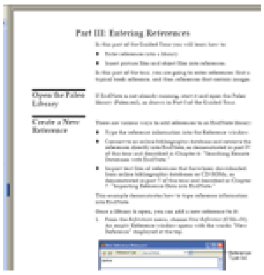
Neilsen provides the following comments:

*"... **the Web is a linking medium** and we know from hypertext theory that writing for interlinked information spaces is different than writing linear flows of text. In fact, George Landow, a Professor of English literature, coined the phrases rhetoric of departure and rhetoric of arrival to indicate the need for both ends of the link to give users some understanding of where they can go as well as why the arrival page is of relevance to them."*

Therefore, we would expect Web writers to split their writing into smaller, coherent pieces to avoid long scrolling pages. Each page would be structured as an inverted pyramid, but the entire work would seem more like a set of pyramids floating in cyberspace than as a traditional "article""

Examples for Narrative

Online news article from the NZ Herald (http://www.nzherald.co.nz)



Page from the EndNote manual showing content list linked to the information

The Mountain Weather forecast from the NZ Met service - http://www.metservice.co.nz/default/index.php?alias=nzmountain

(Based on Pattern 1 'Narrative' from Common Ground – Tidwell 1999)

02 - High-density Information

Context

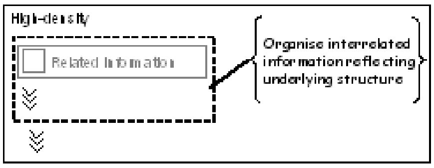
There is a need to convey lots of information, either of a homogeneous type or interrelated in some way, but all of roughly equivalent importance.
Like most patterns that can an overview description of a user interface this pattern can also be used within other patterns such as for information on a surface in a STACK OF WORKING SURFACES.

Problem

In what form should large amounts of often complex information be presented to the user?

Solution

Pack as much information into one working surface as possible, following the precepts of good graphic design, with an organization that accurately reflects the underlying structure of the information.



Reference

Consider providing access methods to the content of the information by using a pattern like COLLECTION BESIDE CONTENT or if very complex CASCADE COLLECTIONS. CONVENIENT ENVIRONMENT ACTIONS may be required where there are actions specific to the type of information being displayed.
Choose a pattern that represents the structure of the underlying information such as HIERARCICAL SET or TABULAR SET. CHOICE FROM A LARGE SET may also be required where the information has a simpler structure as can CHOICE FROM A SMALL SET where displaying all options is possible and desirable. As this pattern is presenting complex information some of that information may require the use of the other high-level patterns such as NARRATIVE or CURRENT PROPERTIES.
Where there is no regular underlying structure to the information being displayed together consider organising it using INFORMATION ON FORM or alternatively where it is just not possible to display all the information on the display surface available group the information using a STACK OF WORKING SURFACES.

Forces

1. The user also wants to quickly get a "big picture" of what the information is about.
2. The user wants to quickly find specific information.
3. The user shouldn't have to do any more thinking, or manipulation of the artifact, than is absolutely necessary.
4. The user may be frustrated, or make a poor decision, by not being able to see all the available information.

Discussion

This is a high-level pattern; you are still left with a decision on exactly how you present the information. How the information is presented depends on what the user is most likely trying to do with it -- whether they need qualitative or quantitative information (or both), for instance, or whether it's more important to find one datum, or to see that datum in context with the rest of the data, or to get a big picture. The answer partly depends upon the information's structure. If the information is geographically structured, a map might be appropriate. Use visual elements to help the user locate different types of information. Data sets of high dimensionality demand more sophisticated display techniques, often leading designers to combine spatial dimensions with colour, and animation. You may need to provide a number of different navigation features so the user can 'wander through' the information space.

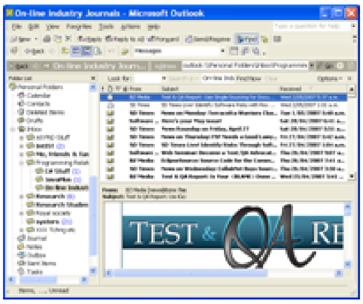
Make the information dense enough so that the eyes do not have to move far from one thing to another and so that scrolling is unnecessary whenever possible. Sparingly use bright colour and/or imagery to highlight specific objects or CURRENT PROPERTIES relevant to the task at hand, so that the user doesn't have to read text or scan linearly to find important things. Use negative ("white") space or subtle colour, rather than boxes or lines, to organize the information. Don't be shy about using large areas, or small fonts, or tiny controls and peripheral things if the information display is the most important task of the artifact.

When a great deal of information is well-organized and presented in a visually reasonable fashion, the human mind is amazingly good at viewing it all, finding specific pieces of information in it, and getting the big picture out of it. Try to take advantage of human cognitive skills to enhance the data display. It's easy to confuse "too much clutter" with "too much information" -- it's likely that most cases of the latter are actually misunderstood cases of the former. If a user complains to you that "There's too much stuff to see at once," see if the problem isn't really that it's just badly presented too many boxes (imagine a 30x10 grid of white edit boxes on a gray background!), too many or too few colours, a hard-to-read font, poor use of negative space or widgetry, etc.

For good discussions of these issues, see Edward Tufte's books, particularly *The Visual Display of Quantitative Information* and *Envisioning Information*. Ben Shneiderman's latest edition of *Designing the User Interface* has an excellent chapter on information visualization, with plentiful examples from the software world.

Examples for High-density Information

NZ Met Service snow conditions at Ski resort centres.



Outlook email message viewer **NZ Met Service local weather reporting**

(Based on Pattern 2 'High-density Information', Common Ground – Tidwell 1999)

03 - Current Properties

Context

The information describing an artefact must be displayed. The information maybe changed over time.

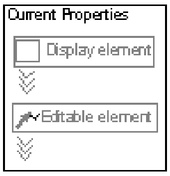
Like most patterns that can an overview description of a user interface, this pattern can also be used for the content part of the pattern COLLECTION BESIDE CONTENT or for information on a surface in a STACK OF WORKING SURFACES and also as part of a display of HIGH-DENSITY INFORMATION.

Problem

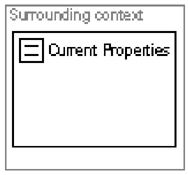
In what form should information that describes the characteristics of something be made available to the user?

Solution

Choose a structure to display the information that fits the type and quantity that needs to be displayed. Put the information describing different characteristics together in a way that emphasizes the important things, deemphasizes the trivial, doesn't hide or obscure anything, and prevents confusing one piece of information with another.



alternatively



Reference

Whether information has to be provided by the user or not use the one of the following patterns that best reflects the structure of the information to be displayed. Where information just needs to be laid out so the user can easily access it, consider using INFORMATION ON A FORM. MASTER WITH DETAILS should be used where the information about an artifact has associated with it many repeated items. Where the information to be displayed is effectively updating in real time consider using a CONTROL PANEL. In situations where there are actions that pertain to all the information such as saving any updates consider including CONVENIENT ENVIRONMENT ACTIONS.

Forces

1. The user wants one place where they know they can find the information about a selected artefact.
2. The information should be organized well enough so that the user can find what they need quickly, interpret it appropriately and modify it if permitted.
3. The information displayed needs to be unobtrusive if it is not critically important.
4. The information does need to be obtrusive if something important happens.

Discussion

The simplest form of current properties is to display one or more items of information, for example, the time. These displays of state information should be unobtrusive and where possible the user should still have control over how, where and when the display occurs. Some current properties change comparatively rarely and may be stored for example in a database especially information associated with business applications. This information needs to be organised so that items of interest can be quickly located and the structure of associated information is clear.

If the user needs or wants to supply values to change the current properties it should be clear how they can do this, for example, it should be easy for the user to quickly find how to change the date. Only dynamically rearrange the displays or controls if the user selects such an option. Call attention to important information by positioning it where the reader expects important information to occur for example those people who embrace a European or American culture tend to read left-

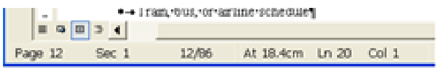
to-right, top-to-bottom, and that something in the upper left corner will be looked at most often.
Visually group together discrete items which form a logical group, and do this at several levels if you have to.
Especially where change in state could be very important, such as the battery of a laptop getting very low in power, use such devices as bright colour, blinking or motion, sound, or all three –to attract the user's attention. But, use a technique appropriate to the actual importance of the situation to the user (such as Important Message).

Examples for Current Properties

Call No.	004.6 Shn
Descript	xviii, 652 p. : col. ill. ; 24 cm
Edition	4th ed
Bibliog.	Includes bibliographical references and index
ISBN	0321197860
Author	Shneiderman, Ben
Subject	Human-computer interaction
	User interfaces (Computer systems)
Other Author	Plaisant, Catherine

Call Number		Location	Status
004.6 Shn		Turitea Book	AVAILABLE
004.6 Shn	c.2	Turitea Book	AVAILABLE
004.6 Shn	c.3	Turitea Book	AVAILABLE
004.6 Shn		Albany Book (Lvl 3)	AVAILABLE
004.6 Shn		Wellington Book	AVAILABLE

Massey University Library Catalogue showing the status of different copies of Ben Shneiderman's book.



State bar from Microsoft Word

Train, bus, or airline schedules such as this one for the Overlander

(Based on Pattern 3 'Status Display from Common Ground – Tidwell 1999)

04 – Information on Form

Context

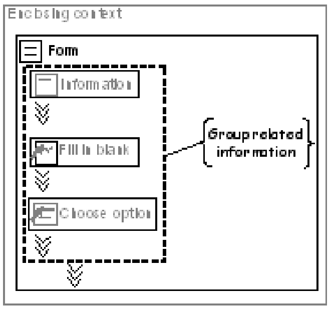
A user often needs to view information that provides a snapshot of something represented by values as in CURRENT PROPERTIES. The user may also have information they need to provide so a task can be completed. The user has normally to enter values as short answers in response to short prompts or questions or by selecting from a set of options. This pattern may be part of HIGH-DENSITY INFORMATION.
This pattern is often required to display the details section found in a MASTER WITH DETAILS pattern.

Problem

How should the application show the current status of the information and indicate what information should if any be supplied by the user?

Solution

Provide appropriate "blanks" to be filled in, which clearly and correctly indicate what information should be provided. Visually indicate those editable blanks consistently, such as with subtle changes in background colour, so that a user can see at a glance what needs to be filled in. Label them with clear, short labels that use terminology familiar to the user; place the labels as close to the blanks as is reasonable. Arrange them all in an order that makes sense semantically, rather than simply grouping things by visual appearance.



Reference

There are a number of basic shapes that information on a form can take. The simplest is just a linear layout of the information items along with editable items. A form will normally have related information organised into GROUPS WITH TITLES alternatively where there is too much information to display on a single surface rather than have the user moving through the information consider using STACK OF WORKING SURFACES to group it. Where all the information naturally organises into rows, consider using TABULAR SET.
For editable items select the pattern that helps the user provide the required data, selecting from CHOICE FROM A SMALL SET, CHOICE FROM A LARGE SET, FORGIVING TEXT ENTRY and ASK FOR INPUT and where possible always provide GOOD DEFAULTS. When the user needs to carry out specific actions provide the required set of CONVENIENT ENVIRONMENTAL ACTIONS, for example, Next and Previous where the form is displaying customer information. When the user has to provide information then it is important that POINTER SHOWS AFFORDANCE is used so the user knows where they are within the group of items.

Forces

1. It should be clear what the user is supposed to read, and what to fill in.
2. The user needs to know what is required, and what is optional.
3. The user needs to know what kind of information to provide.
4. Users almost never read directions.

5. Users generally do not enjoy supplying information this way, and are satisfied by efficiency, clarity, and a lack of mistakes.

Discussion

Visually, arrange the blanks and labels according to a spatial grid; align the edges of the stronger graphic elements, like boxes, where you can. Give them a neat look and a good visual rhythm, but don't let a strong geometry overwhelm the meaning of what is written or provided -- big arrays of edit fields (or whatever) look scary! If there are a lot of them, at least separate them into small group and give them a name according to the meaning of the information if possible. Do not just what group what looks good.

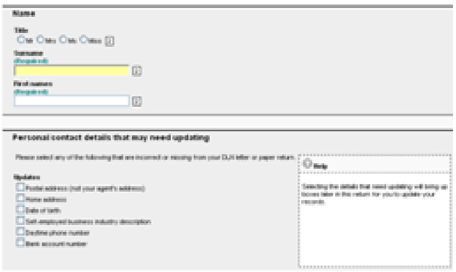
Provide reasonable default values wherever possible, to lessen the amount of work that the user has to do. If the user provides information that makes some parts of the form irrelevant then disable them. Likewise, if the user provides some clue which enables a software-based form to predict what information will be filled in elsewhere, then have the software do it for them. For example, if a user is providing their address but only their base phone number - then fill in their area code for them. But don't do it wrong; that's worse than not doing it at all, because the user then has to both notice the error and correct it.

Be forgiving about the format of what the user provides where possible such as a date try to interpret what they mean. But if the required information simply must follow a certain format, then an empty, featureless text field is the worst possible thing to have to fill in. The user never knows if they're doing it right. Always offer a clue about what is expected by constraining the user's input in a reasonable way, and don't resort to validating afterwards, giving the user a nasty message about how wrong their input was. Giving an example right there on the form is a better idea, but still not as good as a visual constraint. (Sometimes, cultural factors make visual constraints unnecessary. A login screen, for instance, usually just has fields for "User name" and "Password" -- the possible input strings are constrained, of course, but 99.9% of its users will understand exactly what is expected, and constraints or extra information will be more trouble than it's worth. See Don Norman's *The Design of Everyday Things* for lengthy discussions of physical and cultural constraints.)

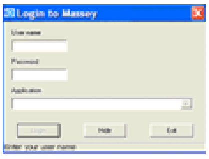
No one ever reads or fills in forms for fun. They do it because they want something (as with a catalog order), or because they are compelled to do it (taxes), or because it's their job (data entry). It's pointless to be cute or clever, and if you make the user have to read extra instructions, or go back to redo something they misunderstood the first time around, or jump around the form to fill in things "out of order," they will be irritated.

If you can find a non-contrived way to use something other than a form, do so. For instance, why ask someone to laboriously fill in a day, month, and year when you can just have them pick a day from a calendar? Or better still fill in today's date if that is the usual value.

Examples for Information on Form



On-line tax form from the www.ird.govt.nz's web site



Simple verification for Massey University's administration systems

(Based on Pattern 4 'Form', Common Ground – Tidwell 1999)

05 - Control Panel

Context

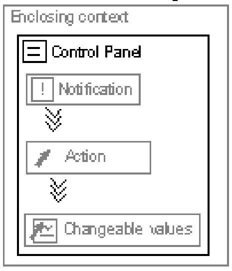
The CURRENT PROPERTIES the user needs to view is the current values of items as they are changing in real time. Often there is also a need to provide a way for the user to either change the state of some values, or command some action take place immediately.

Problem

How best to present the current values the user needs to view and the actions that the user may need to take?

Solution

For each function or state variable that is part of the user's mental model, choose one well-designed control that performs the function or displays the variable's value; put them all together such that the most commonly-used controls are the most prominent.



Reference

Some patterns you can use to create a control panel are CHOICE FROM A SMALL SET, CHOICE FROM A LARGE SET, and SLIDING SCALE where there is a range of values to choose from. Make it clear which controls represent which actions by organising the control features using GROUPS WITH TITLES. When a given control or action is not meant to be used at a given time, disable it. Often a small subset of the commands or actions are the most commonly used so hide the others using EXTRAS ON DEMAND. Some CONTROLS PANEL will require commonly used CONVENIENT ENVIRONMENTAL ACTIONS. Use POINTER SHOWS AFFORDANCE so the user knows where they are on the control panel.
When the control panel is being used to start something then provide GOOD DEFAULTS for editable controls.

Forces

1. The user wants one place where they know they can find the necessary controls for an object, without having to hunt for them.
2. The controls on it should be organized well enough so that the user can find what's needed, with minimal effort and with no confusion.
3. There might be a lot of controls involved, some of which may be complex.
4. The user may need to access a number of the controls in a time period so they want control of how long they stay visible.

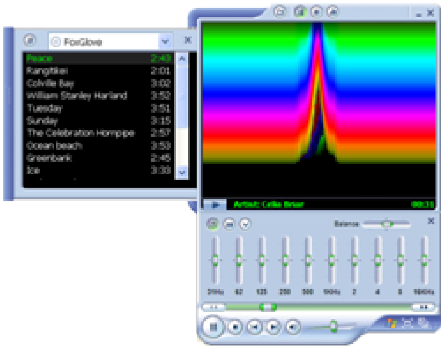
Discussion

Similar functions may have similar-looking controls, but make sure that they aren't so similar that the user gets confused about which is which, even if their labels are different. (Hence the remote controls and cellular phones that are bad examples. They usually have rows of buttons that all feel alike to one's fingers -- and these are the kinds of things that people like to use without being able to look at them!)
When someone uses the controls, give immediate feedback that something is happening; this could be visual feedback, verbal, aural, tactile, etc. The best controls also display the current state since it makes sense to people to affect something in the same place they see it.

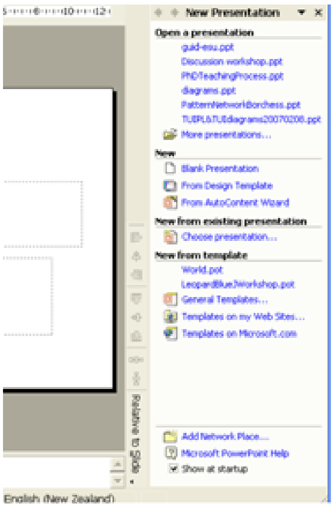
If the things being controlled have obvious and familiar spatial layouts then use that in the control panel. In *The Design of Everyday Things*, Don Norman makes an example out of stoves -- their burners are arranged in a 2x2 grid, but the controls for them are usually not, and users have to stop and think about which goes with which. If analogous spatial arrangement doesn't make sense in a given situation, then try instead to group the controls semantically. Ideally, the controls relevant to a given high-level task will end up clustered together, so the user doesn't have to hunt all over the control panel for the next needed control.

Examples for Control Panel

A Boeing 777 glass cockpit. http://en.wikipedia.org/wiki/Glass_cockpit



Microsoft Media Player Control Panel showing state of current play list



PowerPoint's "New Presentation" task pane provides a panel of related commands

(Based on Pattern 5 'Control Panel' from Common Ground – Tidwell 1999)

06 - Master with Details

Context

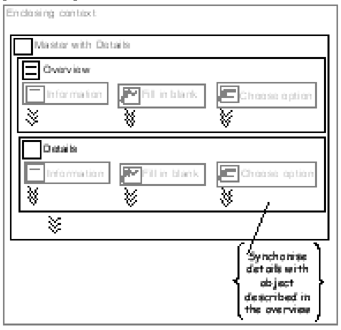
There is a need to convey related content to the user where the content can be clearly partitioned into information that is about a single object which is linked to a number of related items. Use this pattern where each major or master object is associated with many homogeneous items which the user may need to see. Each of these items is described using similar information or components. This pattern is a structural pattern where the user needs to view a snapshot of the related information as in CURRENT PROPERTIES.

Problem

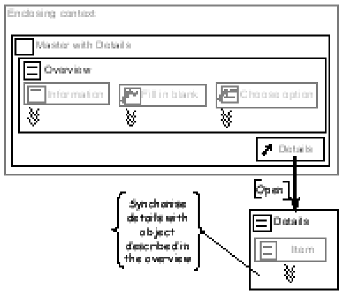
How can an object with a closely associated series of content be presented so that a user can explore the details without loosing sight of the main item?

Solution

Show the basic information about an object as an overview in one part of the display area. Enough details of each of the related items should be available to display in an adjacent area. The display of each of these items should be similar.



alternatively



Reference

Most examples of this pattern use INFORMATION ON FORM to display the information for the master item.

The structure of the details section associated with the overview or master information will dictate the best way to display it. In its simplest form the detail section will be displayed using a simple TABULAR SET with each row displays one of the related items. Another common pattern where there is too much information for each of the items the details could be displayed using a STACK OF WORKING SURFACE. More complex information may call for using a pattern such as COLLECTION BESIDE DETAIL. Consider using EXTRAS ON DEMAND for the details where the user chooses not to view them.

Ensure that appropriate CONVENIENT ENVIRONMENT ACTIONS are provided for actions such as saving and edited content.

Forces

1. The structured information being displayed is inherently a one-to-many relationship.
2. The user is focused on a single main object or 'master' but usually wants to see both it and related 'detail' information simultaneously.
3. The detail information may need to be selectable to show further information about the object.
4. There is not enough space to show multiple detail views
5. It is desirable to make effective use of available screen space by simplify it to a single working surface.

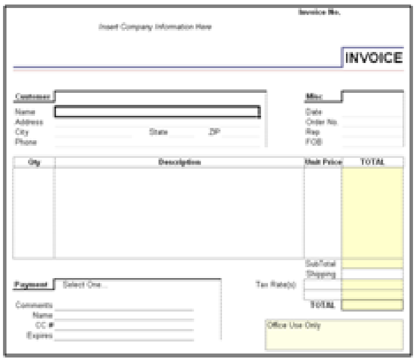
Discussion

Keep the master information and the related detail areas spatially adjacent, so that the user can easily glance back and forth. Generally master information is at the top of the display area, details

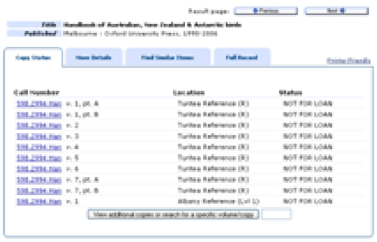
in the centre, and optionally a footer may contain a summary of the details (often calculated values) or possibly actions required to complete or confirm the transaction. Information for the associated items or details is often arranged in rows and columns. Where the number of detail items is large then a technique for display of larger sets such as scrolled regions may need to be used.
The underlying structure of the information is often a hierarchical series of one-to-many relationships between the main object of interest and related detailed objects. A typical use is to display a purchasing transaction with a customer, for example, on an invoice customer details match to a set of invoice line items. This underlying structure needs to be considered when determining how the information is to be arranged on the form.
Often the user needs to be able to select the individual items in turn to view specific details if there is too much detail to show on a single screen.

Where space allows, when the user selects an object, category, or area of interest within the overview, immediately show its related content -- its detail -- in the remaining space. As the user changes the selection, update the detail area to always reflect the current selection.

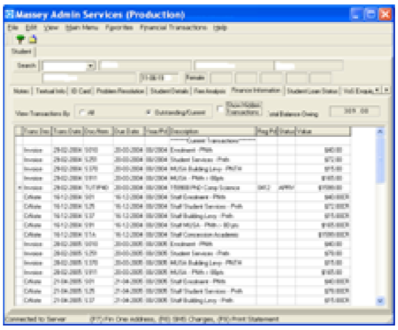
Examples for Master with Details



Template from Microsoft Excel for a typical Invoice



Massey University Library Catalogue entry for a book showing details of individual copies



Tabular form showing financial information for the selected student

(Based on Pattern 4 'Form' & Pattern 9 'Overview beside Detail', Common Ground – Tidwell 1999)

07 - Collection beside Content

Context

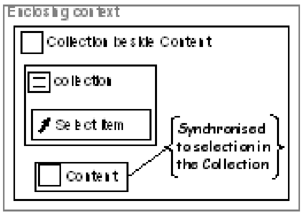
A collection of objects, categories, or even actions needs to be displayed and each of these items has content associated with it that the user wants to access. The user needs to see the overall structure of the collection so they can select items of interest in any order. The pattern high-density information often uses this pattern to partition complex information so it is easier to locate details. NARRATIVE can use this pattern to create a list of contents to provide an alternative access mechanism. The MASTER WITH DETAILS pattern can use this pattern for displaying the details section.
Cascading COLLECTIONS is used for more complex content when the structure of the information has multiple levels of hierarchical information then this pattern will normally be used for the last collection in that structure.

Problem

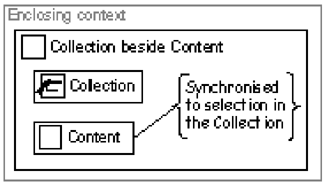
How can you present a large amount of content matched to each member of a set of items so that a user can explore it at their own pace, in a way which is comprehensible and engaging to the user?

Solution

Divide the interface space into two areas. In the first, show the collection or overview from which the user can select at will; in the other, show the content of the selected item.



alternatively



References

This pattern is basically a form of select and show and any of the patterns that display collections from which a selection can be made could be used. The collection should normally be presented using either CHOICE FROM A SMALL SET or CHOICE FROM A LARGE SET depending on the size of the collection. For small linear collections also consider using STACK of working surfaces.
For the content section use one of the high level patterns such as Narrative OR CURRENT PROPERTIES that best represents the overall structure of the content being displayed. Then where necessary select related lower level patterns to fully represent the structure of the content. Where the user can carryout actions on the selected item provide these as a set of CONVENIENT ENVIRONMENT ACTIONS.

Forces

1. The user doesn't want details hidden while selecting the next item to view and visa-versa. Physically, the display area has to be large enough to show parts of both the collection and the content on the workspace.
2. The user needs to be able to find and select items from the collection.
3. The user wants to see where they are in the collection as they review the details of a selected item.
4. The user needs to be able to move easily and quickly between selected items in the collection.
5. The user wants to concentrate on one object, category, or area of interest at a time.

Discussion

In the Western world users are used to using contents lists in written material such as books to quickly scan access information to locate a page number and then finding the associated content. Similarly, in the user interface world placing two panels side by side has become an extremely common and powerful way of presenting identifying information such as titles or subject lines from which a user can select at item of interest and view the contents of that item in an adjacent window. It should be clear to the user which item in the collection has been selected. It is general best to use a technique that is similar to other systems the user normally interacts with to highlight the selected item.
The collection from which the user s will select items should be placed at the top or to the left and the details of the content either below or to the right. Users who read left-to-right languages are trained to expect this type of visual flow. When the user selects an item, immediately show its contents or details in the second

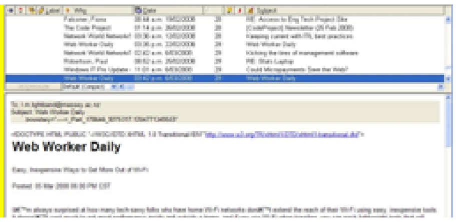
panel. The structure used for displaying the collection should reflect the inherent structure of the items in the collection. Some of the types of structure that are commonly found are

⇒ Linear, usually sorted
⇒ 2D tables, normally Sortable on one of he column headers
⇒ An item that has many related sub-items
⇒ A hierarchy made up of categories and related subcategories
⇒ Spatial organizations, such as maps

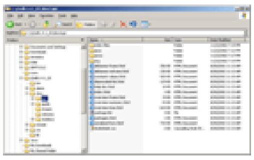
As Tidwell says when both panels are visible side-by-side, users can quickly shift their attention back and forth, looking now at the overall structure of the list ("How many more unread email messages do I have?"), and now at an object's details ("What does this email say?"). This tight integration has several advantages:

⇒ It reduces physical effort. The user's eyes don't have to travel a long distance between the panels, and he can change the selection with a single mouse click or key press, rather than first navigating between windows or screens (which can take an extra mouse click).
⇒ It reduces visual cognitive load. When a window pops to the top, or when a page's contents are completely changed, the user suddenly has to pay more attention to what he's now looking at; when the window stays mostly stable, as in a Two-Panel Selector, the user can focus on the smaller area that did change.
⇒ It reduces the user's memory burden. Think about the email example again: when the user looks at just the text of an email message, there's nothing onscreen to remind him where that message is in the context of his inbox. If he wants to know, he has to remember or navigate back to the list. But if the list is already onscreen, he merely has to look, not remember. The list thus serves as a "You are here" signpost.

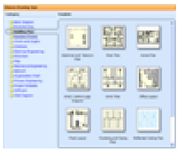
Examples for Collection beside Content



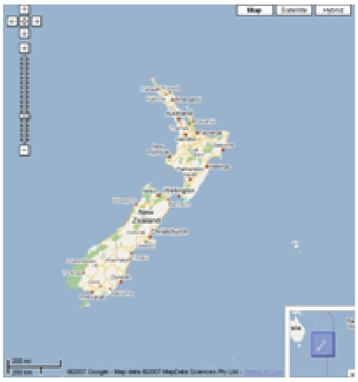
Eudora email message view



From Windows Explorer



From Microsoft Visio – New graphic



From Google Maps

(Based on Pattern 9 'Overview beside Detail', Common Ground – Tidwell 1999 and Pattern 13 'Two-Panel Selector', Designing Interfaces – Tidwell 2006)

08 - Cascading Collections

Context

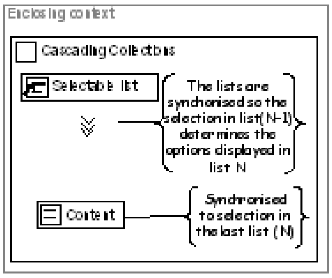
The content to be accessed is inherently hierarchical and multi-layered High-density information. The hierarchy might be deep, and it might have many items on each level.
The hierarchy may be literal, such as a filesystem, or conceptual -- this pattern is often used to let a user navigate and choose items within categories, or make a series of interdependent choices.

Problem

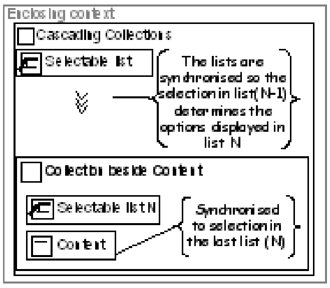
How should series of interdependent items or actions be organized so that they are easy to locate?

Solution

Express a hierarchy by showing selectable collections of the items from each level. Selection of any item shows that item's children in the next collection.



alternatively



Reference

This pattern effectively displays multiple synchronised instances of either CHOICE FROM A LARGE SET representing a deep hierarchical structure. Where there is a limited set of options in a layer then CHOICE FROM A SMALL SET may be more appropriate. The last two panels in this pattern are often similar to COLLECTION BESIDE CONTENT as normally some form of content needs to be displayed.

Forces

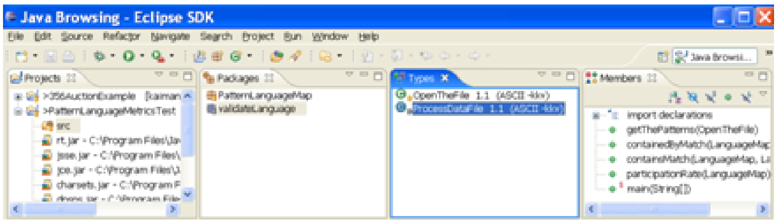
1. The user has to select from many closely related options but the selection of one option restricts the selection of the next set of options.
2. The user needs to get high-level view of all the related sets of options.
3. The options need to be adjacent to each other yet each subset needs to be distinct.
4. The user should be able to easily change an option and not loose other selections unnecessarily.

Discussion

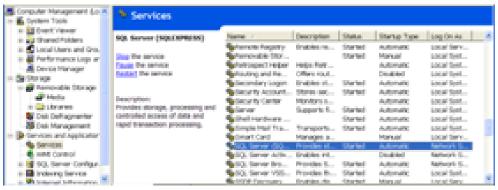
CHOICE FROM A LARGE SET using A Hierarchical Set could work, but the user would need to scroll up and down a lot to see all the items, and wouldn't necessarily get a good overview of the items at higher levels in the hierarchy. By spreading the hierarchy out across several scrolled lists, you show more of it at once. It's that simple. Visibility is helpful when you deal with complex information structures. Also, laying the items out in lists organizes them nicely -- a user can more easily keep track of where he is than he could with an outline format, since the hierarchy levels are in nice, predictable fixed-position lists.
Put the first level of the hierarchy in the leftmost list (which should use single-selection semantics). When the user selects an item in it, show that item's children in the next list to the right. Do the same with the child items in this second list, and show its selected item's children in the third list. Once the user reaches items with no children -- the "leaf" items, as opposed to "branches" -- you might want to show the details of the last-selected item at the far right. Many Finders shows a thumbnail picture to represent an image, or the first few words of a text file; you might instead offer a UI for editing an item, or for reading its content, or whatever is appropriate for a particular application.

A nice thing about this pattern is that you can easily associate buttons with each list: delete the current item, move up, move down, and so on.

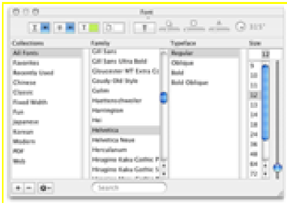
Examples for Cascading Collections



Eclipse showing the Java Browsing view: projects>packages>types>members



Windows Computer Manager showing Services & Applications>Services>SQL Server (SQL Express)



Font dialog box from Mac's TextEdit

(Based on Pattern 63 'Cascading Lists', Designing Interfaces – Tidwell 2006)

09 - Stack of Working Surfaces

Context

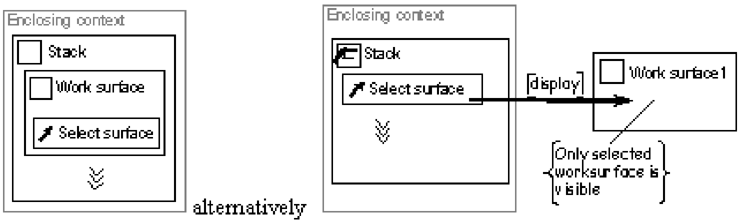
This pattern is often used in conjunction with other patterns like COLLECTION BESIDE DETAIL, INFORMATION ON FORM or MASTER WITH DETAILS. For some complex HIGH-DENSITY INFORMATION this maybe one technique for simplifying the display but it is important to only use this pattern where users don't need to see more than one section of the information at a time.

Problem

When there's too much material on the work area and it is not necessary to have everything on one surface how should the artifact organize the information and actions?

Solution

Stack the surfaces together. Label each surface with a unique and recognizable name or icon (or let the user pick the label), and visually cluster those labels together near the stack. Make sure the user can see which surface is currently selected.



Reference

For each surface use a pattern that visually reflects its underlying structure consider which high level patterns NARRATIVE, HIGH DENSITY DISPLAY or CURRENT PROPERTIES is appropriate. Don't forget to provide CONVENIENT ENVIRONMENT ACTIONS on each work service that requires them. It is also important to use POINTER SHOWS AFFORDANCE to provide the user with feedback on which surface in the stack they are about to select.

Forces

1. The user wants easy access to many working surfaces.
2. There may not be enough space to show them all together.
3. Each surface needs, or could at least use, all the space available.
4. The user can identify them by name or icon, so that they can be brought to the top when needed.
5. Some users don't want to (or can't) manage the working surfaces' positions and sizes themselves.

Discussion

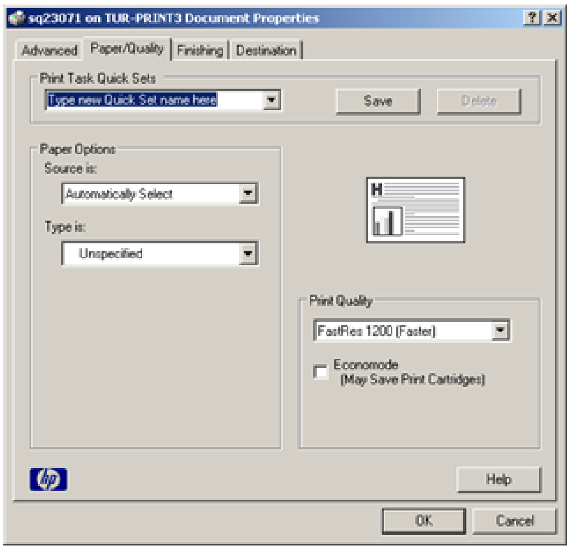
Normally the content can be organised into groups with titles, but there is still too many to all fit on one work area.

This approach can work well for information that doesn't have a very rigid structure. It is important to provide a very simple means by which a user can indicate via the label to bring the selected surface to the top of the pile" such as a touch with a fingertip or a click with a pointer. If the stack is basically static, don't dynamically rearrange the relative positions of the labels, since the user then has to relearn the layout. The labelled "cards" structure the content into easily-digestible chunks, each of which is now understandable at a glance. Also, tabs, the most common form of Card Stack, are very familiar to users.

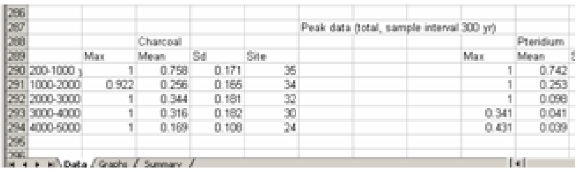
Effective tab pages on dialogs seem to max out at around 10 to a stack, but no one seems to have trouble with, say, an address book with 26 tabs from A to Z. Why not? Is it a matter of knowing exactly what to expect when you get there, or is it the entirely predictable organization of the labels, or is it just easier to deal with paper tabs than virtual ones?

You will also need to find an organizing principle for the working-surface labels. Ask yourself how they relate to each other structurally are they a flat ordered set? - A hierarchy? - A network? - A table?

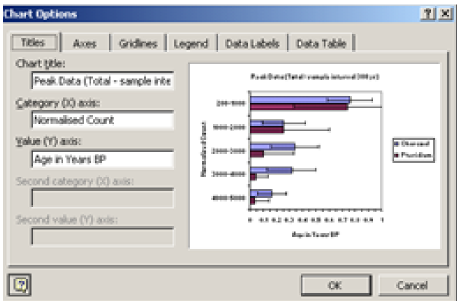
Examples for Stack of Working Surfaces



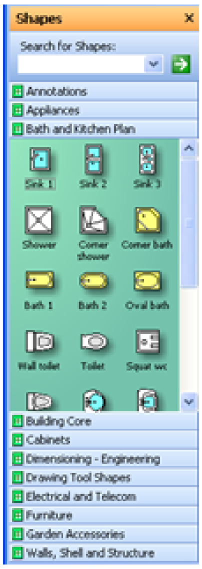
Tabbed pages from printer dialogue



Access to a stack of worksheets in Excel



Organising properties of a graph in Excel



Shapes pallet from Microsoft Visio

(Based on Pattern 24 'Stack of Working Surfaces', Common Ground – Tidwell 1999)

10 - Hierarchical Set

Context

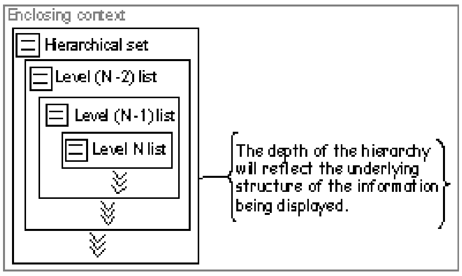
There are many things to show the user, and they are interrelated in a hierarchy (or can be made to appear that way). This could be part of a HIGH-DENSITY INFORMATION display or as the organizing principle within a CHOICE FROM A LARGE SET. It can also be used within cells of complex tables where the overall structure is a TABULAR SET.

Problem

How should the information that is inherently hierarchical be displayed?

Solution

For simple hierarchical data put it in list then use an indented outline structure to illustrate the tree structure.



Reference

I it is essential to help the user navigate their way through the pieces of information so use the pattern POINTER SHOWS AFFORDANCE to indicate where they are in the structure. You get a kind of extras on-demand for free if you let users' open and close parent nodes -- if someone wants to see the children of a given node, they can, or they can ignore it. Where there is a level in the hierarchy that needs to be displayed as multiple columns use TABULAR SET just within that level.

Forces

1. The user should see the structure of the data.
2. Tree like structures are easy to understand.
3. The structure needs to be quick to traverse, when the user knows where something is in the hierarchy.

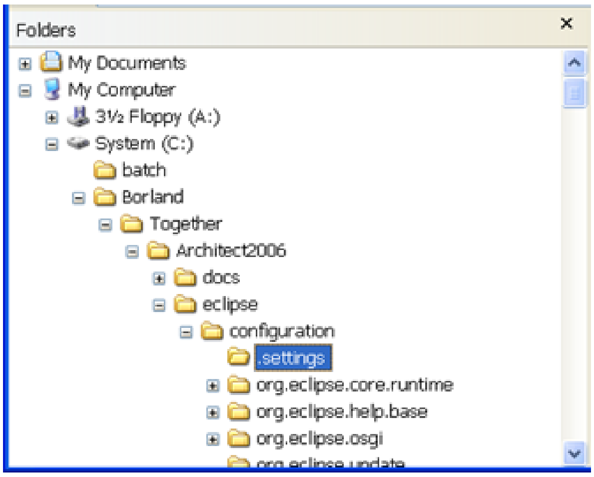
Discussion

A tree like structure is not the only way to display hierarchical information. One alternative to this pattern to consider when deciding which organising principal to use is Cascading Collections.
Keep all the nodes at a given depth from the root in the same line, plane, or arc, to emphasize their parallelism. Allow non-leaf nodes to be opened and closed, to give the user control over how much of the hierarchy is visible at any one time. Depending on expected usage, try to balance the demands of having a dense, fully-visible structure with the ability to look at the details of individual nodes

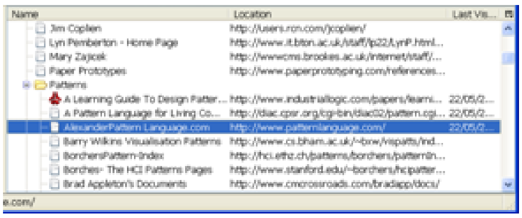
The standard file explorer has brought the columnar "outline view" into common usage in desktop GUIs, and every self-respecting toolkit has one; they're great for fitting a hierarchy into a narrow space, but if you don't have that space restriction, try to use something more visually interesting, like an actual 2D drawn tree. These are much better at showing and manipulating large hierarchies than outlines are, because you don't have to stack everything together in one column, and then scroll forever to get a big picture. Doing them right can be a non-trivial programming problem, unfortunately.

Beware of using trees if the relationships among the data represent a directed graph or network instead, such as with multiple inheritance in a class diagram. It can be done, but it may not be helpful to the user -- they then see one thing in several places in the hierarchy, which may be very confusing. If it's truly a network then use a graph like structure to access the data.

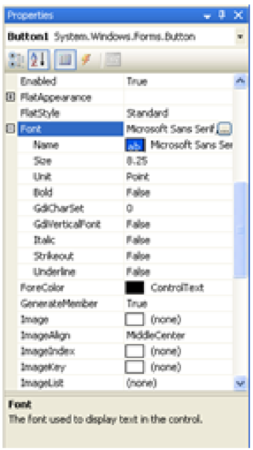
Examples for Hierarchical Set



Directory system viewed using Windows Explorer



Mozilla-Firefox bookmark manger



Visual Studio property sheet for a visual component

(Based on Pattern 12 'Hierarchical Set', Common Ground – Tidwell 1999)

11 - Choice from a Small Set

Context

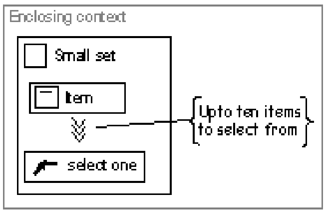
The artifact shows, or allows the user to select or set, a value which is one out of a small set of possible values or items where number (N) is normally ten or fewer. This pattern is used to organise information in many patterns that display properties such as INFORMATION ON FORM or CONTROL PANELS. It may also be found within a HIGH-DENSITY INFORMATION display. It can be used where it is reasonable to display all options so a user can select an item from list like organisation as found in COLLECTION BESIDE CONTENT and CASCADING COLLECTIONS or in conjunction with CHOICE FROM A LARGE SET when the size of the set can vary.

Problem

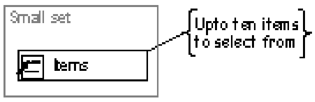
How should the artifact indicate the options from which the user can make a selection?

Solution

Show all the possible choices up front, show clearly which choice(s) have been made, and indicate unequivocally whether one or several values can be chosen



alternatively



Reference

Good Defaults may let the user look at the default value, judge it to be OK, and move on without even bothering to select or set another value. Even though the set is comparatively small it is still important to use POINTER SHOWS AFFORDANCE so the user knows where they are but if the choices are pictorial, or are cryptic in some other way, Short Description may be needed to describe the choices further.

Where the individual pieces of information is a set of individual values then consider using TABULAR SET to display them.

Forces

1. The user should see all the possible values, to put the actual value in context.
2. If the user needs to set the value (not just look at it), they should know what choices are available.
3. Small numbers of things can be taken stock of quickly, and don't take up much space.

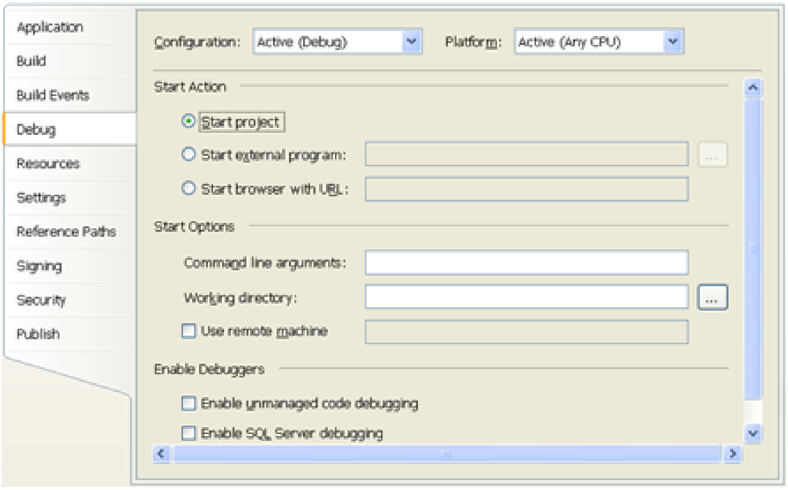
Discussion

This pattern is often one of the patterns used to present values of one or more of the members organised using Groups of Related Things. When the options that need to be displayed gets too large then Choice from a Large Set should be used but be aware that there are some larger sets where the user expects to see all options displayed at once.

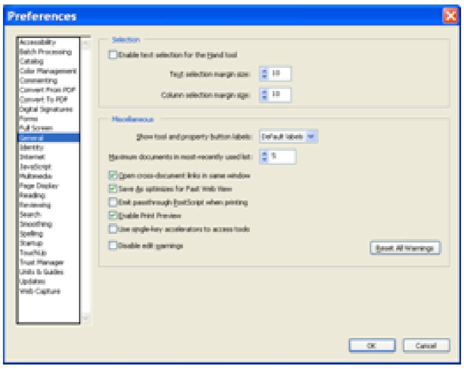
Provide a choice for "Other" or "None of the above," if that will ever be an issue -- don't prevent a user from providing correct information, if they're in a better position to know what's "correct" than you are.

With physical or electronic artifacts (i.e. not paper), a single selection can be enforced by causing the previous choice to "unselect" when the next choice is made. Old car radios did this, and GUI radio boxes do it as well. A user will normally discover and understand this very quickly.

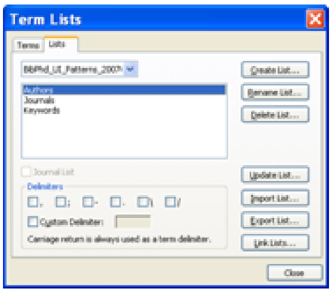
Examples for Choice from a Small Set



Set of radio buttons for selecting required Start Action for a Visual Studio Application



Spinners used for selection the size of margins for Acrobat settings



Simple list of Lists from EndNote

(Based on Pattern 34 'Choice from a Small Set', Common Ground – Tidwell 1999)

12 - Choice from a Large Set

Context

The artifact shows, or allows the user to select or set, a value which is one out of a large set of possible values(normally more than ten) where it is either inappropriate or not possible to display all values in the list at once.

This is a useful pattern to consider when dealing with HIGH-DENSITY INFORMATION. It can also be found within many patterns that display properties such as INFORMATION ON FORM or Control Panels.

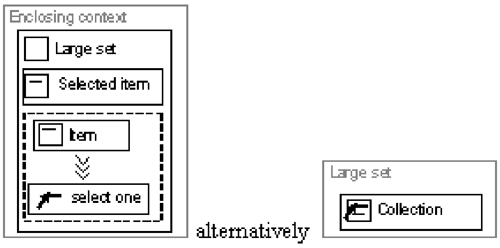
It may also be used in conjunction with more complex list like organisation as found in COLLECTION BESIDE CONTENT and CASCADING COLLECTIONS.

Problem

How should the artifact indicate where to look for the option that the user may want to select?

Solution

Clearly show the selected value up front; organize the set of possible values, but hide them nearby so when not the focus of attention they take up less space.



Reference

GOOD DEFAULTS lets the user see the default value, judge it to be OK, and move on without having to set the value. If the choices are pictorial, or are cryptic in some other way, a SHORT DESCRIPTION may be needed to describe the choices further as well as POINTER SHOWS AFFORDANCE to indicate to the user where they are in the list.

This pattern can be used in conjunction with the more complex lists patterns such as HIERARCHICAL SET and TABULAR SET when the list information is not a set of individual values. It is useful to use this pattern in conjunction with CHOICE FROM A SMALL SET when the number of values in the list can vary. In such a situation the display moves from using CHOICE FROM A SMALL SET to CHOICE FROM A LARGE SET when the list no longer fits the available space and visa versa as the list shrinks.

Forces

1. The user should on demand be able to see as many values as possible, to put a selected value in context.
2. If the user needs to set the value (not just look at it), they should know what choices are available.
3. The user should be able to find the value they want quickly.
4. Large numbers of things take a long time to read and take up lots of space.

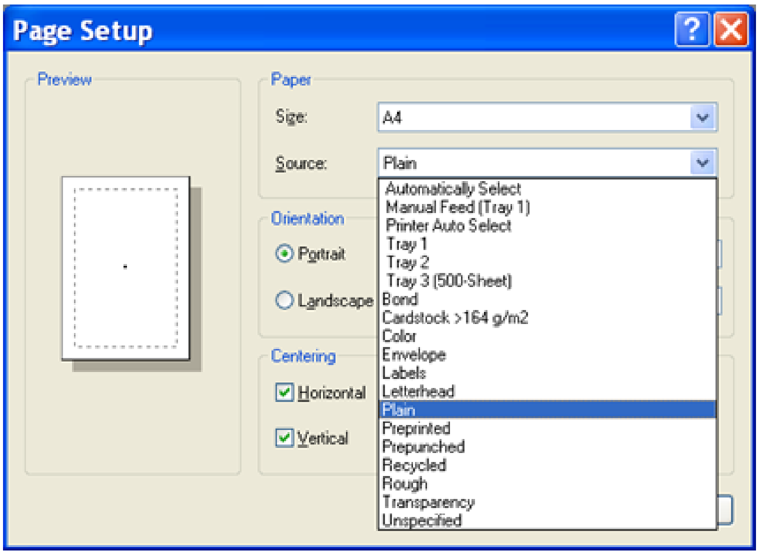
Discussion

This pattern is used when is not appropriate or practical to display all the values in the list all the time. It is for list structures where the user only needs to see some values and then only when the list is the focus of their attention.

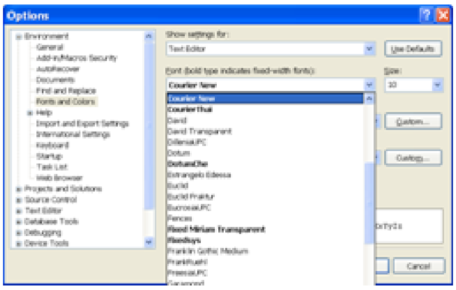
Put the items in the list on a separate working surface, for instance, or in a scrolled area or combo box in a GUI environment, where they are only a single gesture away. Organize them in the way most appropriate to how the user will be searching -- alphabetically if looking for names, numerically for a font size, most-often or most-recently used for a document to edit, etc. Allow a user who knows exactly what they want to directly enter the choice, as by typing, rather than by laboriously searching it out.

Scrolled combo boxes are really only necessary if the dropdown list is going to run off the edge of the screen -- it's easy to miss the choices beyond the visible area, and it's awkward for many people to drop down the list, then move to the scroll bar (or buttons), then scroll up, then down, etc. It takes the bad features of scrolled lists and makes them worse; first by making you show the list and then by shrinking the scrollbar.

Examples for Choice from a Large Set



Ordered drop-down list from Paint showing most common first.



An ordered scrolled hierarchical list box and an ordered scrollable drop down list from Visual Studios Options dialog.

(Based on Pattern 35 'Choice from a Large Set', Common Ground – Tidwell 1999)

13 - Tabular Set

Context

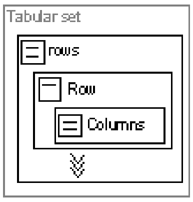
This is often used as part of a High-density Information DISPLAY OR SIMPLE PRESENTATION OF A TABLE OF DATA REPRESENTING PROPERTIES displayed by INFORMATION ON FORM. Where there is overview information that matches many items that can be displayed using a tabular format this table becomes part of the Master with Details pattern.
Where information within a table needs to be selectable then consider using this pattern within a CHOICE FROM A LARGE SET or CHOICE FROM A SMALL SET depending on the amount of data. This pattern can also be used to display some levels within information that has primarily a hierarchical structure as a HIERARCHICAL SET.

Problem

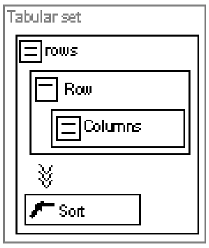
How should multivariate information be organized so the user can explore, reorder, customize, search through for a single item, or simply understand on the basis of the different variables?

Solution

Show the data in a table. In most cases let the user sort the table rows according to the column values.



alternatively



Reference

When it is not practical to provide features so that the columns can be manipulated (e.g. sortable, or with changeable widths) use Pointer Shows Affordance to help the user see where they are in that table structure. With complex data the first column may use hierarchical set. The content found in each of the cells of a tabular structure may be dependent on the context using this pattern.
Where the user can provide information as a changed value for one or more items then use patterns that help the user provide the correct information. Use such patterns as FORGIVING TEXT ENTRY or ASK FOR INPUT and where possible always provide GOOD DEFAULTS

Forces

1. The user should see the structure of the data.
2. The user wants to easily look up a specific piece of information.
3. Putting similar information into columns facilitates quick comparison of values.
4. It's easy, and intuitively obvious, to sort a table according to the data in the columns.

Discussion

Order it according to some appropriate organizing principle, either by the value of some column especially if the table's principal use is to compare items by that value; or alphabetically by item when the principal use is to look up values. It is often enough to put some white space between columns to set them apart, but not too much; the user's eye shouldn't have to work too hard to go from one column to another.
The columns themselves should be organized logically. Depending upon the data and the user's purpose, the columns may be organized with the most commonly needed data immediately after the item name, and in decreasing order of importance as you move from left to right. Alternatively, they may be organized into groups, with the group titles in a line above the column names.
Giving the user the ability to change the sorting order of a table has powerful effects. First, it facilitates exploration. A user can now learn things from the data that he may never have seen

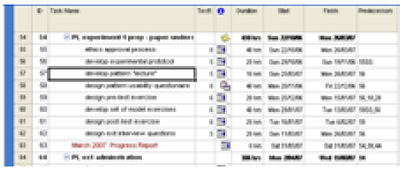
otherwise -- How many of this kind? What proportion of this to that? Is there only one of these? What's first or last? Suddenly finding specific items becomes easier, too; a user need only remember one attribute of the item in question (e.g. its last-edited date), sort on that attribute, and look up the value he remembers.

Furthermore, if the sort order is retained from one invocation of the software to another, the user now can effectively customize the UI for his preferred usage patterns. Some users want the table sorted first to last; some last to first; and some by a variable no one else thinks is interesting. It's good to give users that kind of control.

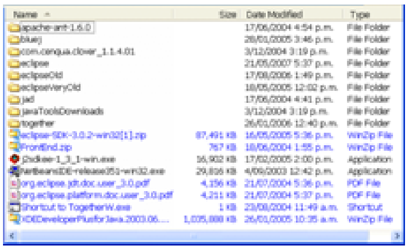
Finally, the clickable-header idiom has become familiar to many users now, and they may expect it.

Examples for Tabular Set

Catalogue of Equipment for sale at http://www.heavy-machinery.co.nz



Task list from Microsoft Project



The detailed file list view from Windows Explorer

(Based on Pattern 13 'Tabular Set', Common Ground – Tidwell 1999)

14 - Convenient Environment Actions

Context

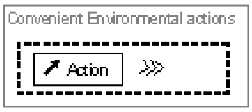
The user can take an action that affects the existence or state of the whole or selected artifact. This pattern is ubiquitous and can be used with many patterns that display information in NARRATIVE form, as HIGH DENSITY INFORMATION. Often more than one set of CONVENIENT ENVIRONMENT ACTIONS can be required in different parts of the overall display. Multiple instances are often required for completing INFORMATION ON FORM, CONTROL PANEL or MASTER WITH DETAILS patterns. This pattern is also required when actions need to be carried out on the selected content displayed using COLLECTION BESIDE CONTENT.

Problem

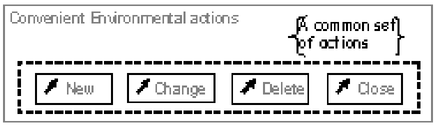
How should the artifact present the standard actions the user would expect?

Solution

Group the widgets for the actions (e.g. buttons or menu items) together, label them with words or pictures whose meanings are unmistakable, and put them where the user can easily find them regardless of the current state of the artifact.



alternative



Reference

Both Good Defaults and POINTER SHOWS AFFORDANCE are often used with this pattern.

Forces

1. The user should know exactly how to stop or leave this artifact at any time.
2. The user should know what other actions are available.
3. The user may already know what they have to do, but they need to find the corresponding action.
4. The user may need to perform these in a hurry, or under stress.
5. Doing these actions accidentally may be disastrous.

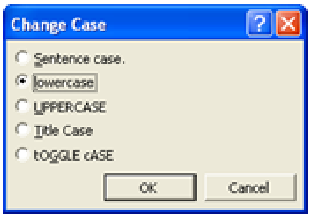
Discussion

Use their design and location to make them impossible to confuse with anything else. Set them up so they are not easy to trip accidentally -- hardware devices can use physical barriers to do this, but software's more difficult. Confirmation dialogs are clumsy but somewhat effective; use them until someone invents something better. If a state change or a shut down might cause a disaster, then at least try to make the effects reversible. Disable state-change actions whenever they become irrelevant or impossible DISABLED IRRELEVANT THINGS, but never do this to controls that close or quit the artifact.

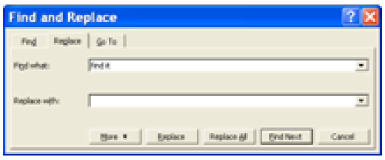
Is this whole pattern just a consequence of clumsy design in the first place? Something about the "books and charts" comment in the Context makes me think about the necessity of explicit Environment Actions in software and electronics. If we could design these so well that they no longer need explicit commands for on/off, save state, change mode, etc., then we might get truly dramatic gains in usability. I don't know how to do this, of course. If I did, I would be off designing things that way, not writing this pattern.

Icon buttons are often used for these. As long as familiar symbols are used, they work well; use the standards for the artifact's intended domain and culture. ("X" for quit, "?" for help, check-mark for OK, and 0/1 for off/on are common ones.) When the icons for these basic functions are incomprehensible, or if you hide them in an unexpected place, the user has to carry a heavier memory burden.

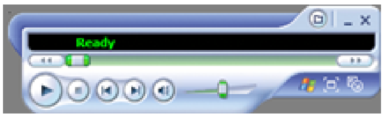
Examples for Convenient Environment Actions



OK & Cancel buttons on dialogs



Standard action buttons on a Microsoft search dialog



Start, pause or resume, stop, go-to-start, go-to-stop. on a multi-media control

Full Screen/Minimize/Close controls for a window

Back/Forward/Undo/Stop/Home buttons on a Browser

(Based on Pattern 31 'Convenient Environment Actions', Common Ground – Tidwell 1999)

15 - Pointer Shows Affordance

Context

Patterns that tend to use this a lot are interactive ones where the user is selecting parts like in choice from a small set OR choice from a large set. This pattern indicates which item in a display is being 'pointed at' as the user moves over the display. It is important to use this pattern with any pattern where the user will be searching a group of similarly constructed values such as in hierarchical set or tabular set.

It is also appropriate to use this pattern when information has been grouped in some way such as found in INFORMATION ON FORM, CONTROL PANEL and STACK OF WORKING SURFACES.

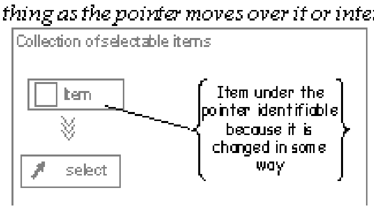
This pattern is used in conjunction with CONVENIENT ENVIRONMENT ACTIONS where the user is trying to select an appropriate action to take.

Problem

How can an action that the user takes be shown using a visual representation?

Solution

Change the affordance of the thing as the pointer moves over it or interacts with it.



Reference

A pattern that is ubiquitous for most user interfaces where the user interacts using some form of pointing device.

Forces

1. Static visual affordances aren't always enough to indicate the presence of a control that can be manipulated, especially when space is tight or when an elegant-looking graphic design is of paramount importance.
2. Multiple affordances can be more effective than one, if they work together properly.
3. If the pointer is near the border of a control that can be manipulated, especially something with a small target area, the user should get feedback that instantly tells them whether they're "in" or "out."

Description

Don Norman brought the term "affordance" into the interface designer's vocabulary with his classic *The Design of Everyday Things*. In it, he defines an affordance as "the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used." On a UI this can be done in one of two ways - by changing the pointer to a small picture illustrating what can be done, or by changing the thing itself to make it stand out visually.

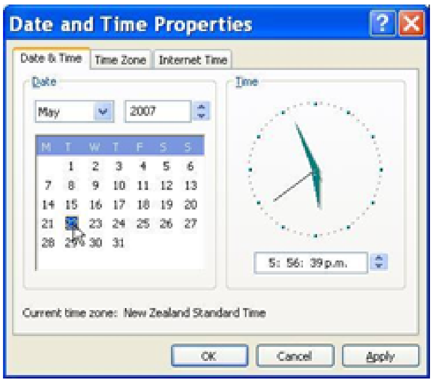
If you change the pointer, use a small picture illustrating what can be done. Use a standard icon if an appropriate one can be found -- crosshairs for drawing, single arrow for selection, I-beam for text entry, hands, pencils, paintbrushes, resize arrows, etc. -- because they are so easily recognized. Keep it small or mostly transparent, so that the user can easily see what's under it.

If you change the thing itself, you have a lot of freedom to experiment. Any visual change may be enough to tell a user that the object is at least clickable; but consider your audience when deciding how flashy or distracting the change is. To be sure that your design actually works, of course, you should test it with potential users.

Be careful not to use this pattern gratuitously. Now that the tools to implement it are widely available, lots of user interfaces use it as a substitute for static visual affordances. This isn't always

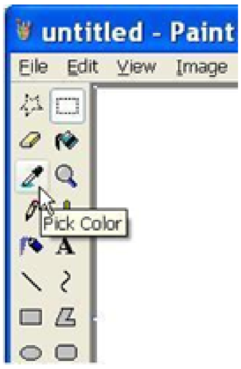
wise. Think about the poor user looking at a screenful of borderless icons, some of which are buttons, some of which are moveable objects, and some of which don't do anything at all! .
For those of us stuck with non-tactile interfaces, such as mice, this pattern produces something like a substitute tactile sense. As you run the pointer over the interface, you get visual responses that correlate to physical sensations -- bumpiness (raised button edges), heat (when something turns from a muted colour to a bright colour), etc.

Examples for Pointer Shows Affordance

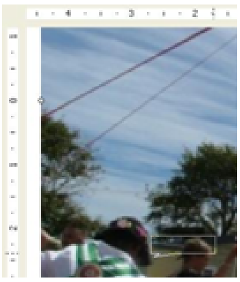


Day highlights as user moves mouse over it

Arrow pointer indicating resizing of area selected



Tool 'popping out' from tool bar as mouse-over occurs



Cross hairs showing positioning of callout.

(Based on Pattern 17 'Pointer Show Affordance', Common Ground – Tidwell 1999)

16 - Groups with Titles

Context

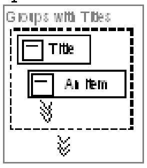
There's a lot of content to show the user and it is possible to organise the content into logical groupings.
This pattern is used to complete higher-level patterns including (but not limited to), INFORMATION ON FORM OR Control Panel.

Problem

How should the items or actions be organized so that they are easy to scan and easy to locate?

Solution

Group the closely-related items together with a visually strong title. Identify the groups by separating each visually often using white space.



References

This pattern is more like a general principle that you will apply when it is appropriate. Use it to group together related information that occurs in many other patterns.

Forces

1. Large, undifferentiated masses of things can be intimidating and difficult to figure out, especially to someone seeing them for the first time.
2. The brain is very good at discerning groups and nested groups of things, and at getting a "big picture" from the grouping structure.
3. The items cannot or should not be shown on separate working surfaces; for instance, because they are part of one coherent task.
4. People naturally assume semantic coherence where there is visual coherence.

Discussion

If the content is too large to fit on a single artefact then consider how the groupings can be organised into a Stack of Working Surfaces which is an alternative way of grouping items and actions under a title. If a group of items requires a set of associated actions such as SAVE, NEW etc. then this pattern is not appropriate use something that better reflects the information such as INFORMATION ON FORM or even CONTROL PANEL.
When the user sees content sectioned neatly into chunks, their eye is guided along more comfortably. The human visual system always looks for bigger patterns, whether they're deliberate or not. So put them in deliberately! Group those items that are more closely related to each other than other things for example by organising into thematic or task-based sections that will make sense to the user.
Give each section a useful title. Well-defined and well-named sections structure the content into easily-digestible chunks, each of which is now understandable at a glance. It makes the information architecture obvious. (Information architecture is basically about rendering content in a way that communicates its semantic structure.)
It is preferable to keep the number of things in any one group to around ten or fewer. This is a very basic way of managing complexity, and is almost more of a principle than a pattern. The "ten or fewer" comes from Miller's number (7+-2), which represents, among other things, the upper limit of someone's ability to "instantly" scan a set of items. Beyond that, the time it takes to read through the items grows linearly. Use repetition and symmetry to keep the groups from becoming visually chaotic (and don't overuse boxes, even the nicely etched ones in some GUI toolkits; white space often works just as well). Make sure that the grouping is not arbitrary, but is based on the meaning of what is being shown -- as pointed out in the Forces, a user will naturally try to derive some kind of semantic meaning from the groupings, even if it's wrong.
To make a visual grouping of things look good, it's tempting to short-change their individual usability. Try not to make too many sacrifices here. For instance, a common mistake is to make a text box too short for the

expected input, simply to make it visually fit in. Also, there's no need to be dogmatic about this pattern -- expert users for instance, may prefer that all the available actions are as densely packed as possible, to save space, and don't require titles.

Sometimes a large group of homogeneous items can work together to form a single conceptual entity. This is true of a lot of data rich content -- data points on a scatter chart or a column of numbers representing a single variable. This pattern shouldn't apply to them. See Edward Tufte's book *Envisioning Information,* in particular the chapter on "Small Multiples," for an excellent set of counterexamples these large sets of items are meant to show small changes between individual items that share most characteristics, to make those changes stand out. The impact comes from seeing all those similar items next to each other. In many cases, that impact would be completely lost if you broke up those sets of items into small groups!

When applying this pattern, get the information architecture right -- split up the content into coherent chunks, and give them short, memorable names (one or two words, if possible). Next, choose a presentation:

⇒ For titles, use a **font that stands out** from the rest of the content -- bold, wider, larger point size, stronger colour, etc. (Remember that nothing's stronger than black, not even red.)

⇒ Try **reversing the title** against a strip of contrasting colour. White on dark can make it look like a Windows title bar.

⇒ Use **whitespace** to separate sections.

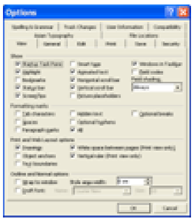
⇒ Putting sections on **different background colours** works well on Web pages and "flashy" interfaces, though it's unusual on desktop UIs.

⇒ **Boxes** made from etched, bevelled, or raised lines are familiar on desktop user interfaces. They can get lost -- and just become visual noise -- if they're too big, or too close to each other, or deeply nested. It can be done well when combined with the title.

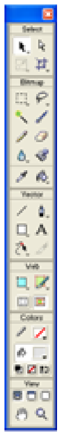
Examples for Groups with Titles

Groups of items and tools for helping New Zealanders managing their money

From a Javadoc HTML page



Typical Options Dialogue from Microsoft Windows



Toolbox from FireWorks MX

(Based on Pattern 11 'Small Groups of Related Things', Common Ground – Tidwell 1999)

17 - Forgiving text entry

Context

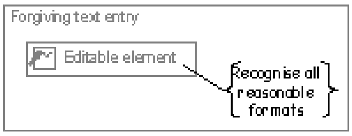
The user should enter information, as required by a number of other patterns such as INFORMATION ON FORM, or TABULAR SET, but that information may be formatted in any one of several ways. The UI asks for data that users might type with an unpredictable mix of white-space, hyphens, abbreviations, or capitalizations. Often, the UI can accept input of various kinds from the user -- different formats, or syntax. But you want to keep the interface visually simple.

Problem

How does the artifact help the user supply valid formatted information?

Solution

Allow the user to enter information into a field using any reasonable format or provide a question prompt that indicates possible options use search action to locate the required input value.



Reference

This can help simplify the UI tremendously, in terms of how many brain cells a user has to use to figure it out. It may even remove the requirement to provide a Short Description, though they're often seen together. Good Defaults may let the user look at the default value, judge it to be OK, and move on without even bothering to set the value; a good default can also help suggest what kind of input is allowed. .

Forces

1. Physical and cultural constraints on the allowable input prevent unnecessary interpretation errors from being made.
2. Too much constraint can be annoying to the user.
3. The user doesn't need the cognitive burden of figuring out what specific format is acceptable.
4. Users are diverse, and will likely have different cultural or personal preferences for formatting various pieces of information (especially things like names, dates, and times).
5. Both computers and humans -- depending on who's interpreting the entered information -- are good at figuring out what a given text entry is intended to be, if it even vaguely resembles some accepted format.

Discussion

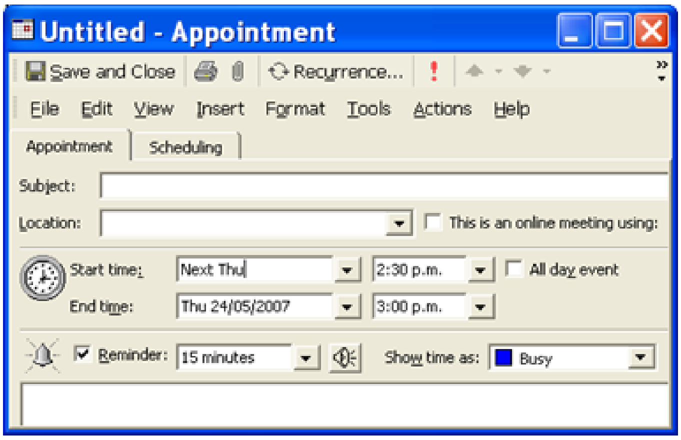
The user just wants to get something done, not think about "correct" formats and complex UIs. Computers are good at figuring out how to handle input of different types (up to a point, anyway). It's a perfect match: let the user type whatever he needs, and if it's reasonable, make the software do the right thing with it.

You might consider STRUCTURED FORMAT as an alternative, but that pattern works best when the input format is entirely predictable (and usually numeric, like telephone numbers). Another alternative to consider is ask for Input WHICH CAN BE USED WHERE IT IS POSSIBLE FOR THE APPLICATION TO DETERMINE THE REQUIRED INPUT FROM THE VALUES THE USER PROVIDES

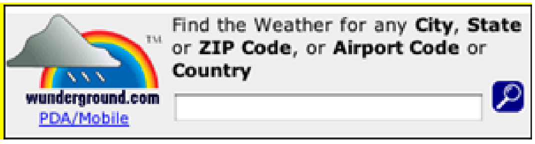
The catch (you knew there would be one): it turns a UI design problem into a programming problem. You have to think about what kinds of text a user is likely to type in. Maybe you ask for a date or time, and only the format varies -- that's an easy case. Or maybe you ask for search terms, and the variation is in what the software *does* with the data. That's harder. Can the software disambiguate one case from another? How?

Each application uses this pattern differently. Just make sure that the software's response to various input formats matches what users expect it to do. Test, test, and test again with real users.
Divide it into multiple fields with different relative sizes, for instance, or superimpose a faint visual design on it (like dividers or decimal points). Be careful not to constrict the input so much that it makes things too complicated, or so that it no longer fits the possible input values that users may need to give it! Do user testing as needed to judge whether or not it's too annoying.

Examples for Forgiving text entry



Entering the date and time into an Outlook Appointment



From http://wunderground.com

(Based on Pattern 38 'Forgiving Text Entry', Common Ground – Tidwell 1999)

18 - Ask for Input

Context

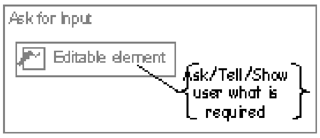
Any INFORMATION ON FORM that requires the user to provide input could use this pattern as can TABULAR SET. Use this where the user needs a short reminder of what is required and a reasonable default can not be provided.

Problem

How does the artifact indicate what kind of information should be supplied?

Solution

Pre-fill a field or editable space with a prompt that tells the user what to do or type.



Reference

This pattern should only be used when the prompting label cannot carry the same information or an illustrative default can not be used.

Forces

1. The user may have no idea of as to what data to enter.
2. The user may not know the type of format that is required.
3. Values for a field may be impossible for the application to determine.
4. The user may not have access to explanatory documentation.

Discussion

Use this pattern where you can't provide a good default VALUE -- perhaps there is no reasonable default. This pattern is an alternative to using a Short Description AND IS USEFUL IN A SITUATION WHERE THE PROMPTING INFORMATION CAN BE DISPLAYED WITHIN THE TEXT ENTRY SPACE. It helps make the UI self-explanatory without the user having to explicitly 'look' for that information by moving the selecting device. ASK FOR INPUT is a sneaky way of supplying help information for controls whose purpose or format may not be immediately clear.

With an Input Hint, someone quickly scanning the UI can easily ignore the hint (or miss it entirely). Sometimes this is your desired outcome. But an Input Prompt sits right where the user will type, so it can't be ignored. The advantage here is that the user doesn't have to guess whether she has to deal with this control or not -- the control itself tells her she does. (Remember that users don't fill out forms for fun -- they'll do as little as needed to finish up and get out of there.) A question or an imperative "Fill me in!" is likely to be noticed.

An interesting side effect of this pattern is that users may not even bother to read the label that prefixes the text field. Often an input prompt includes the words on the prompting label which then becomes almost superfluous, in terms of the form's meaning. Because the user's eye will be drawn first to the white text fields, those prompts will probably be read before the labels anyway! That said, don't remove the labels -- that prompt is gone forever once the user types over it, and on subsequent readings of the form, they may not remember what the control asks for originally.

When choosing an appropriate prompt string, consider beginning with one of these words:

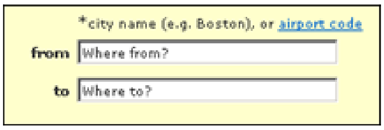
⇒ For a dropdown list, use *Select, Choose,* or *Pick.*

⇒ For a text field, use *Type* or *Enter.*

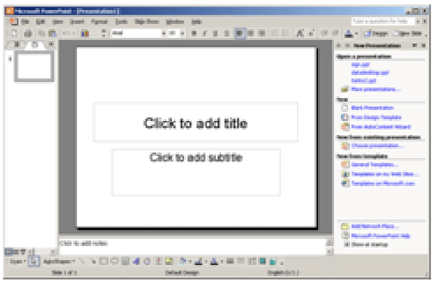
Another good practice is to end the prompt with a noun describing what the input is, such as "Choose a state," "Type your message here," or "Enter the patient's name."

Since the point of the exercise was to tell the user what they were required to do before proceeding, don't let the operation proceed until they've done it! As long as the prompt is still sitting untouched in the control, disable the button (or other device) that lets the user finish this part of the operation. That way, you won't have to throw an error message at the user.

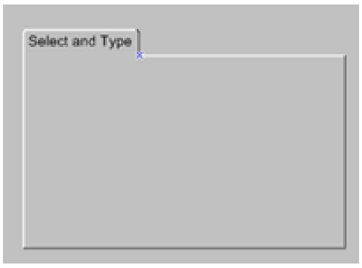
Examples for Ask for Input



The questions guide the user, From http://orbitz.com



Creating a new PowerPoint presentation the user just clicks on area and types to add a new Title



Labelling the tab on a tab widget when creating a UI model using Visio

(Based on Pattern 72 'Input Prompt', Designing Interfaces – Tidwell 2006)

19 - Good Defaults

Context

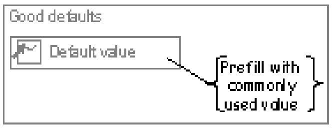
Use in patterns that require the user to provide input like INFORMATION ON FORM, CONTROL PANEL or TABULAR SET. The user has to fill in information or change the setting of a status variable, and the data fields can be given reasonable default values. This can also happen within many other patterns such as - Choice from a Small Set, Choice from a Large Set, convenient Environmental Actions, Forgiving Text Entry, etc.

Problem

How does the artifact reduce the amount of data the user has to provide?

Solution

Rather that letting a user enter information into a blank and featureless textfield, wherever appropriate, pre-fill fields with reasonable default values, preferable from determining user needs.



Reference

Just about every pattern where the user has to type in some value could use this pattern but it can also be used where the user is selecting an item from a collection or even viewing a larger collection – for example, having a map display NZ rather than USA when the email address has the suffix '.nz'.

Forces

1. Filling out forms is not inherently a fun activity; don't prolong the agony by making the user do unnecessary work.
2. The user may have no clue as to what kind of value to supply, from the given context.
3. The user may be perfectly happy with the default behaviour or values, with no desire to change it; but they may want to know what the default values are.
4. "Correct" values for some unfilled fields may be difficult or impossible for the artifact itself to figure out.
5. If the user sees an example of what is expected of them, they don't have to be uncertain about what they're entering into the text field.

Discussion

Your UI asks the user any questions requiring form-like input (such as text fields or radio buttons), and you want to reduce the amount of work that users have to do. Perhaps most users will answer in a certain way, or the user has already provided enough contextual information for the UI to make an accurate guess. For technical or semi-relevant questions, maybe he can't be expected to know or care about the answer, and "whatever the system decides" is okay.

But supplying defaults is not always wise when answers might be sensitive or politically charged, such as passwords, gender, or citizenship. Making assumptions like that, or pre-filling fields with data you should be careful with, can make users uncomfortable or angry. (And for the love of all that is good in the world, don't leave "Please send me advertising email" checkboxes checked by default!)

By providing reasonable default answers to questions, you save the users work. It's really that simple. You spare the user the effort of thinking about, or typing, the answer. Filling in forms is never fun, but if this pattern halves the time it takes him to work through it, he'll be grateful.

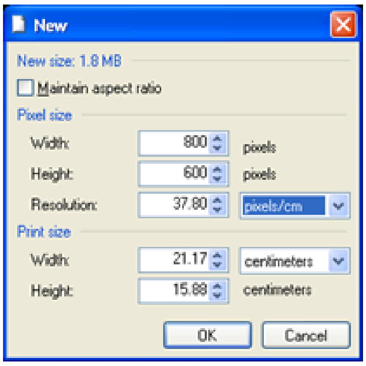
Even if the default isn't what the user wants, at least you offered an example of what kind of answer is asked for. That alone can save him a few seconds of thought -- or, worse, an error message.

Sometimes you may run into an unintended consequence of Good Defaults. If a user can skip over a field, that question may not "register" mentally with him. He may forget that it was asked; he may

not understand the implications of the question, or of the default value. The act of typing an answer, selecting a value, or clicking a button forces the user to address the issue consciously, and that can be important if you want the user to learn the application effectively.
Pre-fill the text fields, combo boxes, and other controls with reasonable default values. You could do this when you show the page to the user for the first time, or you could use the information the user supplies early in the application to dynamically set later values. (For instance, if someone supplies a United States ZIP code, you can infer the state, country, and sometimes the city or town from just that number.)
Don't choose a default value just because you think you shouldn't leave any blank controls. Do so only when you're reasonably sure that most users, most of the time, won't change it -- otherwise, you will create extra work for everybody. Know your users!
Occasional-use interfaces like software installers deserve a special note. You should ask users for some technical information, like the location of the install, in case they want to customize it. But 90 percent of users probably won't. And they won't care where you install it, either -- it's just not important to them. So it's perfectly reasonable to supply a default location.
Jeff Johnson discusses this issue at length in *Web Bloopers: 60 Common Web Design Mistakes and How To Avoid Them* (Morgan Kaufman). He provides some hilarious examples of poor default values.

Examples for Good Defaults

A profile dialogue for personal setting preferences as in Visual Studio



Creating a new picture in Paint.NET

(Based on Pattern 47 'Good Defaults', Common Ground – Tidwell 1999)

Appendix A9: Detail of Recommended Methods from TUIPL

The **T**eaching oriented **UI** design guided by a **P**attern **L**anguage framework (TUIPL) for the UI detailed design stage is composed of five key processes, two conceptual models, a domain-specific pattern language with lo-fidelity and hi-fidelity prototypes as shown in Figure A9.1 below.

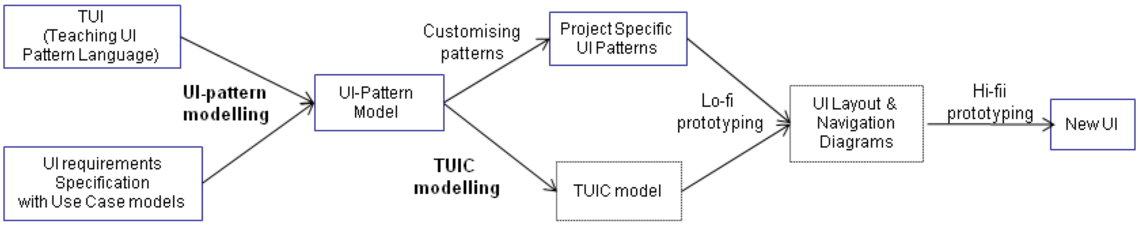


Figure A9.1 - The pattern-guided UI detailed design framework

UI-pattern modelling takes the information defined in the UI requirements and uses this information to select relevant patterns from TUI. These patterns are then used to create the model by selecting relevant links between the patterns and structuring the graph hierarchically.

TUIC modelling brings together canonical abstract prototypes (Constantine 2003a) and navigation diagrams (Constantine 1998) to create TUIC model diagrams illustrating the solutions of patterns making the UI-pattern model. These diagrams are then used to guide the development of the TUIC model for the UI.

Customising patterns takes the information from the UI patterns used in the UI-pattern model and modifies them to refer explicitly to the context and the new UI being modelled. This could include using relevant examples and modifying the TUIC model diagrams to use naming relevant to the UI rather than generalisations.

Lo-fi prototyping produces the UI layout and navigation diagrams, guided by the project-specific UI patterns. The lo-fi prototype is created by restructuring the TUIC Model.

Hi-fi prototyping translates the lo-fi prototype into the final user interface.

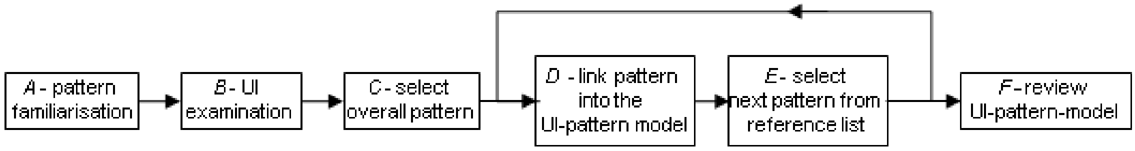
The two methods trialled in this research are the UI-pattern modelling and TUIC modelling. Two types of method were trialled. The first introductory versions were used to introduce the students to the methods and to UI patterns. The second were used for developing the same type of model for a new UI from requirements.

The introductory methods are based on modelling an existing UI. The versions here have been modified to reflect the experience gained from this research. It is suggested that a teacher wishing to use the teaching material may want to update the methods used in the teaching materials to these versions.

The second method types are more flexible than the introductory versions and are recommended for guiding the development of a new UI after students have gained experience with the simplified versions.

A9.1 Simplified UI-pattern Modelling (existing UI)

The modified but still simplified method designed to introduce students to both the UI-pattern models and using UI patterns. It is expected that students will modify the method by introducing intermediate representations (e.g. draft, marking the UI, manipulating the UI-patterns into a hierarchy, marking list of pattern names).

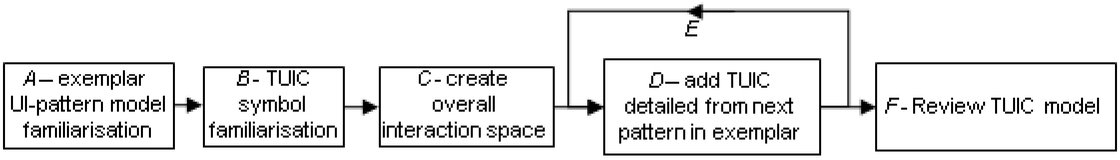


Step A. Become familiar with the subset of UI patterns.
Step B. Examine the user interface.
Step C. Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information the user interface (UI) deals with.
Step D. Link each selected pattern into the UI-pattern model by connecting it with lines showing how it relates to other patterns. When appropriate add a label to the line to match the interaction spaces identified on the UI.
Step E. From the list of patterns that a pattern references, select those that best describe some other feature of the UI. Repeat Steps D and E until no more patterns can be selected.
Step F. Use the context and reference sections of your patterns to check for missing links or patterns.

Figure A9.2- Method for creating a UI-pattern model for an existing UI

A9.2 Simplified TUIC Modelling (existing UI)

The modified but still simplified method to introduce students to using selected UI patterns to guide the development of a TUIC model.

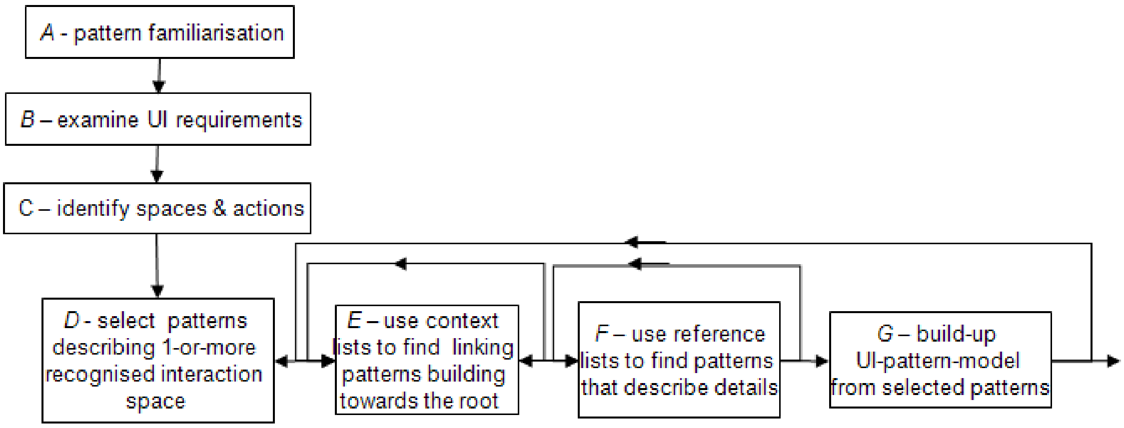


Step A. Find the subset of UI patterns used in the UI-pattern model and become familiar with how they describe the example UI.
Step B. Become familiar with the TUIC components.
Step C. Starting with the top pattern, examine the solution and the TUIC model diagram for an overview of the model you are to create. Draw an outline space and label it.
Step D. Work down the UI-pattern model using each pattern's TUIC model diagram to guide your modelling. Use generic CAP symbols and names when necessary.
Step E. Use lower level patterns to add details and navigation components. Refine the model by using specific CAP symbols and domain-specific labels.
Step F. Review TUIC model and update labels to reflect UI domain

Figure A9.3- Method for creating a TUIC model for an existing UI

A9.3 UI-pattern Modelling (new UI)

A flexible UI-pattern modelling method for creating a model for a new UI.

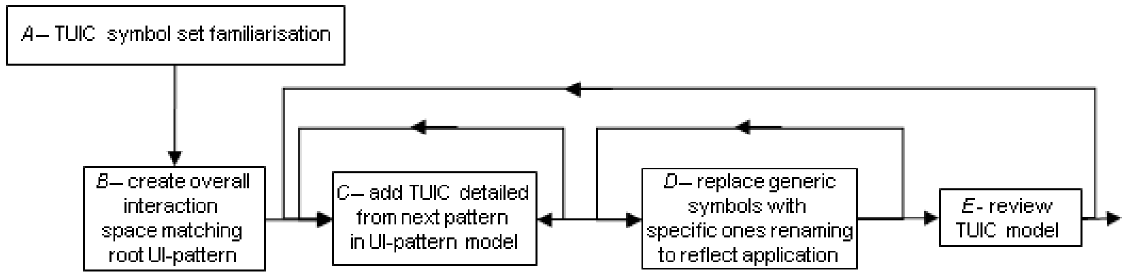


Step A. Become familiar with the UI patterns.
Step B. Examine the UI requirements documentation.
Step C. Identify actions and interaction spaces then group related interaction spaces and any associated actions.
Step D. Select patterns that seem to best describe one or more of the interactions spaces.
Step E. Check each pattern's Context section and select a pattern from the list that gives the best overall description. This description may include patterns already selected. A pattern may be mentioned in the context section of one or more of the selected patterns. Use the context section to find common patterns until one of these higher level patterns can be identified as the root (1, 2 or 3) of the UI-pattern model
Step F. Check each pattern's Reference section to identify patterns that best describe the details and additional features required for the user interface.
Step G. Using selected patterns, create a diagram consisting of patterns connected with lines to indicate how they are linked into a structure. Use the context and reference sections of patterns to check for missing links or patterns.

Figure A9.4- The UI-pattern model building method

A9.4 TUIC Modelling (new UI)

A flexible TUIC modelling method for creating a model for a new UI.



Step A. Become familiar with the TUIC symbol set.
Step B. Starting with the top or root pattern of the UI-pattern model, examine the solution and the TUIC model diagram to get an overview of the TUIC model to be created. Draw the main interaction space and label it.
Step C. Work down through the patterns using the diagram to guide the development of the TUIC model. Add CAP & navigation symbols to define details. In the initial stages use the generic components.
Step D. As the modelling progresses, replace the generic components with specific ones where they make the intent of the design clearer. Rename interaction spaces and components to reflect the application's domain. Active material components should be used to replace equivalent material and tool components. Comments that clarify patterns used should be added.
Step E. Review the model. This review may remove unnecessary layers of nested containers, group and reorder related symbols, and rename symbols. Comments can be added to make types of interactions and requirements clearer. Consider adding content details too.

Figure A9.5- The pattern informed TUIC Modelling method

Appendix A10: Observation Criteria

A10.1 Observation criteria – researcher

Verbal references to patterns by
- Name
- Pattern Id
- Example
- Diagram
- Reference
- Context
- Other

References to parts of patterns either
- Verbally
- Physical – e.g. pointing

Observed physical searching for a pattern by shuffling through the patterns in the collection.
Manipulating patterns into some logical order.
References to parts of the UI illustration as possibly being described by a pattern.
- Verbal
- Physical - pointing

Comments about the process being used
Requests for help
Other

A10.2 Observation Items in the images

Artefacts that are visible
UI Instructions Exercise Sheet Original collection *(still in paper clip)* Pattern list Patterns on list marked Pile of patterns Selected patterns laid out overlapping Selected patterns laid out as graph Draft graph Partial graph Completed pattern graph
Artefacts held or pointed at
Focus on a pattern Placing pattern onto pile Pointing to component on UI Pointing to pattern name on list Pointing to the reference section Pointing to illustration Pointing to the forces section Pointing to the solution section Pointing to the context section Pointing to the discussion section Pointing to instructions

Appendix A11: The Patterns Questionnaire

A11.1 Instructions

Please tick the column that indicates your response to each statement.

		Agree	Undecided	Disagree
	UI – User Interface			

A11.2 Statements

Common to Studies One, Two and Three

1	The information presented in these UI patterns is quite clear.			
2	These UI patterns are helpful when modelling user interfaces.			
3	Usually I found it straight forward to find which pattern to select next.			
4	I enjoyed using these patterns to build a UI pattern model of the user interface.			
5	It was not difficult to find suitable patterns to model the user interface.			
6	Most of the time I could locate the information I needed in these patterns.			
7	The content described in these patterns is very informative.			
10	It is relatively easy to move from one pattern to the next one when selecting patterns to describe a user interface.			
11	I felt in command of these patterns when I used them to create the UI-pattern model.			
13	I would not like to use these UI-patterns when I design a new user interface.			
14	Creating the UI-pattern model could be performed in a straight forward manner using these UI patterns.			
16	There is sufficient information in each pattern.			
18	The UI patterns are easy to understand.			
20	The patterns selected to model the user interface described it quite well.			
21	The organisation of the information in each of these patterns seems quite logical			
22	Learning how to use these patterns was difficult.			
23	There are too many steps required to create the UI pattern model.			
24	The set of patterns provided was not adequate for modelling the UI in the exercise.			
25	It is easy to remember the information in these UI patterns.			
28	Creating the UI-pattern model was not easy			
30	The patterns help me focus my thoughts about building the UI-pattern model.			

Common to Studies One and Two

9	There is never enough information in a single pattern.			
12	These UI patterns are difficult to understand.			
15	Using these UI-patterns helped me discuss UI concepts with my partner			
17	I repeatedly referred back to the same patterns.			
27	The patterns were not helpful when discussing the modelling exercise with my partner.			

Additional Statement Used in Study One

8	Working with these patterns is mentally stimulating.			
19	Using the UI-patterns made me feel more confident about my knowledge of User Interface design.			
26	It is easy to forget what information is recorded in these patterns.			
29	I had to seek assistance to understand how to use these patterns.			

Additional Statements Used in Studies Two and Three

8	The diagram helped me identify the essential parts of a pattern's solution.			
19	I found the pattern names confusing.			
26	The pattern names indicated each pattern's intent.			
29	The examples helped me understand the information in these UI patterns.			
31	I can understand and act on the information provided in these patterns.			
32	I could not understand the diagram illustrating the solution.			

Additional Statements Used in Study Three

9	The patterns helped me focus when building the TUIC model.			
12	Creating the TIUC model using CAP and navigation symbols was relatively easy.			
15	The method for creating an TUIC model has too many steps.			
17	I enjoyed using these patterns when building the TUIC model.			
27	Building the TUIC model could be performed in a straight forward manner using the UI patterns.			

A11.3 Statement Groups

A subset of the statements has been classified into one of six topics. These topics correspond to objectives for the questionnaire.

UI-pattern Model Building

S-14	Straight forward method
S-23	Number of steps OK
S-28	Method easy to follow

Discussion aids

S-15	Aid UI-concepts discussion
S-27	Aid modelling exercise discussion

Pattern Language Structure

S-03	Straight forward to select pattern
S-05	Finding suitable patterns OK
S-10	Easy to find next pattern
S-17	Referring back to patterns

Information in UI patterns

S-01	Clear information
S-07	Patterns informative
S-09	Sufficient information
S-18	Patterns easy to understand
S-21	Logical organisation
S-25	Content easy to remember
S-27	Understand and act on information

Illustrations

S-08	Diagram helpful
S-29	Examples helpful
S-32	Diagram understandable

Patterns names

S-19	Pattern names clear
S-26	Pattern names indicate intent

Appendix A12: Exit Questionnaire

Each of the questions was placed on the form with sufficient space between so the participant could write their answers.

Glossary

UI (User interface) is any interface a user may use to access information or activities using an electronic device including web based systems or sites.

A12.1 Section A - Common to Studies One and Two: Background Information

Circle age range: under **20** / between **20 to 30** / over **30** Circle gender: M / F

1 Which Degree Program (e.g. BA, BSc, BE, DipInfSci) are you currently enrolled in?

2 Please give details of any additional activities including work experience involving user interfaces that may be relevant to your knowledge of user interface development:

3 How proficient do you consider yourself in designing user interfaces?

4 Did you know anything about canonical abstract prototypes before doing this tutorial? - Y/N

If yes, please indicate what you knew and how proficient you think you were in using this method for creating conceptual UI models.

A12.2 Section A - for Study Three: Background Information

1 Name

2 Approximately how many years since you first began designing user interfaces

3 Briefly describe the UI development approaches you currently use or have used:

4 List any other UI design approaches you are familiar with

5 Please outline any other experiences or information that may be relevant (e.g. UI development tools):

6 How proficient do you consider yourself in designing user interfaces?

7 Do you have any knowledge of canonical abstract prototypes? - Y/N

If yes, what is your experience in using them?

8 How proficient do you consider you are in creating abstract prototypes?

A12.3 Section B - Common to Studies One, Two and Three: The Patterns

The different parts of a pattern are:

Identifier - Name
Context
Problem
Forces
Solution
Diagram
Reference
Discussion
Examples

1 Which part or parts of the patterns were the most useful?

Why?

2 Which part or parts of the patterns were the least useful?

Why?

4 Please add any other comments you may have about using the UI patterns when creating the UI-pattern model?

Alternatives for Question 3: The Patterns

Study One

3 Please indicate which of the pattern presentations you preferred:

illustrated / not illustrated (narrative only)

Why?

Studies Two and Three

The pattern form included the diagram in the solutions section which was added to the list of the parts of a pattern.

3 Please indicate whether the diagram helped you identify the essential parts of the solution. – Y/N

Why?

Additional Question Study Three

4b Please add any other comments you may have about using the UI patterns when creating the TUIC models?

A12.4 Section C - Common to Studies One, Two and Three: Creating a UI-pattern model

Listed below are the steps we suggested you use to create the UI-pattern model.

[The steps varied depending on the study as the method was modified based on findings from the previous study. Details can be found in the appropriate section of the thesis]

1 Are there any steps that you consider need to be modified? - Y/N

If yes, what would you modify?

Step **Modification**

Why?

2 Are there additional steps or activities that you consider should be included in the method? - Y/N

If yes, what do you think was missing?

Step **Missing activity**

Why?

4 Please add any other comments you may have about creating a UI-pattern model?

Alternatives for Question 3: Creating a UI-pattern model

Studies One and Two

3 Indicate how you rate creating a UI-pattern model of an existing user interface for developing your understanding of the UI patterns:

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

Why?

Study Three

3 Indicate how you rate developing a UI-pattern model as part of the process when developing a new user interface:

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

Why?

A12.5 Section D - Common to Studies Two and Three: Creating a TUIC model

Listed below are the steps we suggested you use to create the TUIC model.

(The steps varied depending on the study as the method was modified based on findings from the previous study. Details can be found in the appropriate section of the thesis]

1 Are there any steps that you consider need to be modified? - Y/N

If yes, what would you modify?

Step **Modification**

Why?

2 Are there additional steps or activities that you consider should be included in the method? - Y/N

If yes, what do you think was missing?

Step **Missing activity**

Why?

3 In the second exercise you were asked to create a UI-conceptual model of the user interface using TUIC modelling. Indicate how useful the patterns were in helping you learn how to do this?

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

Why?

4 Please add any other comments you may have about creating a conceptual UI model?

A12.6 Section E - Common to Studies Two and Three: Communication aid

1 Indicate what impact the patterns had on how you communicated with your partner as you completed the modelling exercises?

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

Why?

A12.7 Section F - for Study Three: Communication and Design

The user interface patterns provide a common language that professionals and non-professionals alike should be able to understand thereby providing a common language. In particular they should aid users when discussing UI concepts and ideas. Using patterns as part of a user-centered interface design process where users are integrated into the design team should improve communication between team members when designing an interface.

1. Indicate what you consider the impact of using patterns might have on the quality of communication between the professional and non-professionals on a UI design team?

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

   Why?

2. Indicate what impact developing a UI-pattern model might have in improving the quality of communication between members of a UI design team?

1	2	3	4	5
unhelpful	not very helpful	helpful	very helpful	extremely helpful

   Why?

3. Please add any other comments you may have about the impact of using patterns or UI-pattern modelling could have on communication:

4. Please add any other comments you may have about the impact of using patterns or UI-pattern modelling could have in assisting non-professional users work within a UI design team:

A12.8 Section G - for Study Three: Communication and Collaboration

Abstract prototyping uses CAP symbols is used by UI professionals with clients when designing new user interfaces after completing task modelling. TUIC models combine CAP symbols and navigation symbols. UI-pattern modelling can be used as a bridge between a task model and TUIC modelling. It should help a user-centered design team in their discussion and creation of UI conceptual models.

1. Indicate what you consider the impact of using patterns might have on the quality of communication between members of a design team while collaborating on the development of a TUIC model:

1	**2**	**3**	**4**	**5**
unhelpful	not very helpful	helpful	very helpful	extremely helpful

   Why?

2. Please indicate your opinion of the usefulness or otherwise of TUIC modelling as a suitable UI conceptual modelling technique:

1	**2**	**3**	**4**	**5**
unhelpful	not very helpful	helpful	very helpful	extremely helpful

   Why?

3. Please add any other comments you may have about the impact of using patterns or TUIC models could have in assisting non-professional users work within a UI design team:

Appendix A13: TUIC Test

Page 1

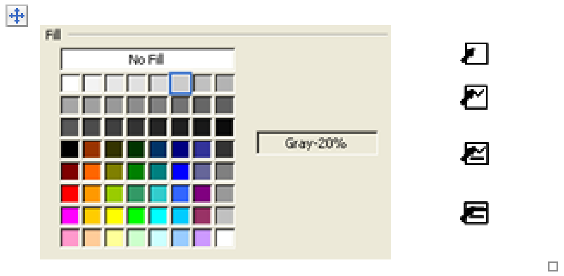
Using UI Patterns to Guide the Development of Conceptual User Interface Models

THE TUIC COMPONENTS QUESTIONS

Canonical Abstract Prototype - CAP

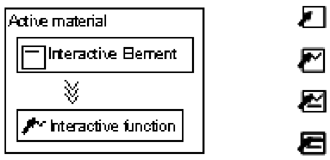
1. Examine the following background element from Word's 'Borders and Shading dialogue's Shading tab.

Circle the active material that best represents it.



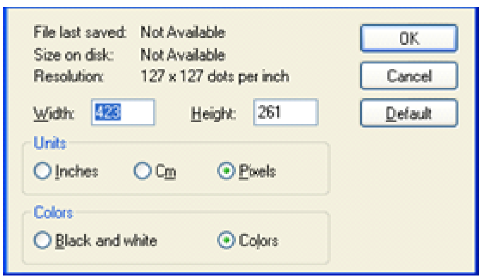
2. There are three types of CAP symbols – tools, materials and active materials. Active materials can be represented using a combination of tools and materials providing a short hand component for common elements found on user interfaces. Examine the following generic example which uses just tools and materials CAP symbols.

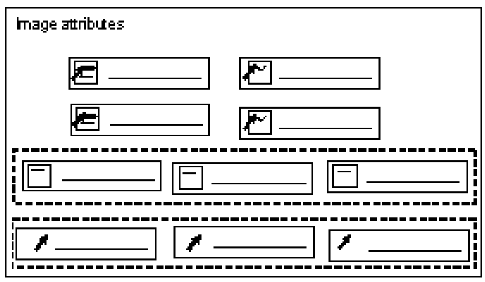
Circle the active material that best represents this generic example.



3. When providing names on CAP symbols that are part of a conceptual model for an interface for a specific application those names should be relevant to the application and the UI element they are modelling rather than generalised as used in the diagrams representing the pattern solutions. Examine the following screen dump and the UI conceptual model that represents it

On the line next to each symbol name the component on the CAP model to indicate the UI element it represents.

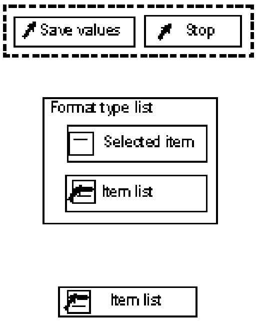


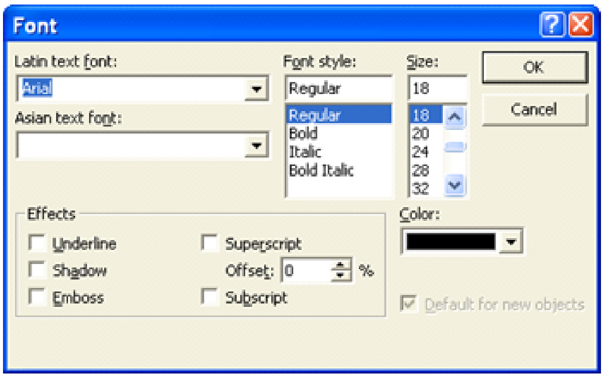


Page 2

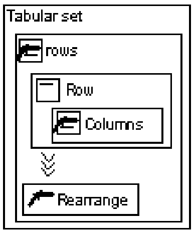
4. Below are three generic CAP symbols and a screen dump of PowerPoint's font format dialog window.

Draw a line from each CAP symbol to one UI element in the dialog window it best represents.

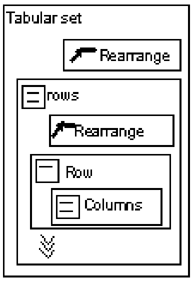


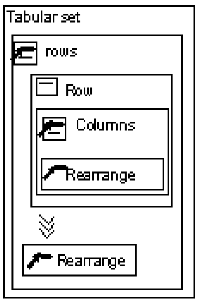


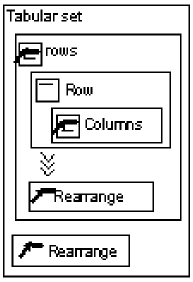
5. The CAP symbol used to display the TABULAR SET pattern is replicated below. It represents a table where the user can only change the order of the rows. Many applications allow the user to also change the order of the columns.



Circle the CAP symbol that best represents a table where the user can change the order of both the rows and the columns.







Appendix A14: Subset of TUIC Components

Subset of CAP Symbols[2]

TOOLS

Symbol	Interactive Function	Examples
	generic action/operation	
	select	Object selector, Group member picker
	modify	Change shipping address, Edit client details

MATERIALS

Symbol	Interactive Element	Examples
	generic container	Any interaction context or space
	element	Student ID, Product thumbnail image
	collection	Personal address, Electrical components
!	notification	Email delivery failure notice, Controller status

ACTIVE MATERIALS

Symbol	Interactive Function with Element	Examples
	generic active material	
	selectable collection	Performance choices, Font selection
	editable element	Patient name, Next appointment date
	editable collection	Patient details, Test object properties

Comments in curly brackets used to clarify behaviour characteristics.

Dashed lines used to distinguish related groups of components – can be a panel or a page.

Triple chevron to indicate that there could be zero to many repetitions of a component.

Subset of Navigation Symbols[1]

Types of transition for moving between interaction spaces.

TRANSITION TYPES

Symbol	Transition
	Transition between context triggered by an action
	Transition with implied return

[Action label] Label in square brackets used to indicate action that triggered the transition

[2] Reference: **Constantine and Lockwood, Ltd**
http://www.foruse.com/articles

Appendix A15: Marking and Grading the Models

A15.1 UI-pattern Models

The models were marked by counting the number of appropriate patterns selected, the number of inappropriate patterns and the number of missed patterns. Because there are appropriate alternatives for the models ratios were used for comparisons.

Appropriate patterns = count of appropriate patterns / (count of appropriate patterns + count of inappropriate patterns)

A similarly ratio for links was calculated. This was based on the patterns the student has selected so that they weren't doubly penalized.

Appropriate links = count of appropriate links / (count of appropriate links + count of inappropriate links)

The second metric was calculated using the count of missed patterns or links.

Potential patterns = count of appropriate patterns / (count of appropriate patterns + count of missed patterns)

Potential links = count of appropriate links / (count of appropriate links + count of missed links)

A15.2 TUIC Models

The models were marked by counting the number of appropriate TUIC components selected, the number of inappropriate CAP components and the number of missed TUIC components. Because there are appropriate alternatives for the models ratios were used for comparisons.

Appropriate TUIC components = count of appropriate TUIC components / (count of appropriate TUIC components + count of inappropriate TUIC components)

Potential TUIC components = count of appropriate TUIC components / (count of appropriate TUIC components + count of missed TUIC components)

The percentage of correct labels used a count of the domain-specific labels and the generic labels.

Appropriate labels = count of domain-specific labels / (count of domain-specific labels + count of generic labels)

Correct nesting was a simple ratio as all alternatives identified required the same level of nesting for a given model.

Correct nesting = nesting count / constant for the model

A15.3 Grading System for Models

The Grading system used is:

Passing grade is 50%.

Of the students who pass:

- No more than fifty percent of students should gain a C grade score, ranging from 50% – 60%.
- No more than twenty percent of students should gain an A grade, ranging from 75% to 100%
- For a B grade marks range from 60% to 75%

Appendix A16: Study One – Numeric Data

A16.1 UI-pattern Model Data

Partitioned by Pattern Form

	Appropriate patterns		Potential patterns		Appropriate links		Potential links		Overall	
	illustrated	narrative	illustrated	narrative	illustrated	narrative	illustrated	narrative	illustrated	narrative
	0.71	0.43	0.44	0.33	0.19	0.13	0.21	0.07	0.39	0.24
	0.71	0.43	0.44	0.33	0.19	0.13	0.21	0.07	0.39	0.24
	0.77	0.64	0.45	0.44	0.50	0.42	0.21	0.21	0.48	0.43
	0.77	0.64	0.45	0.45	0.50	0.42	0.21	0.21	0.48	0.43
	0.80	0.67	0.56	0.45	0.50	0.50	0.29	0.43	0.54	0.51
	0.80	0.78	0.56	0.56	0.50	0.50	0.29	0.43	0.54	0.57
	1.00	0.78	0.56	0.64	0.57	0.60	0.29	0.43	0.60	0.61
	1.00	0.83	0.56	0.64	0.57	0.60	0.29	0.43	0.60	0.62
	1.00	0.86	0.64	0.67	0.60	0.64	0.36	0.50	0.65	0.67
	1.00	0.86	0.64	0.67	0.60	0.75	0.36	0.57	0.65	0.71
	1.00	1.00	0.73	0.73	0.71	0.75	0.43	0.71	0.72	0.80
	1.00	1.00	0.73	0.73	0.71	0.80	0.43	0.71	0.72	0.81
	1.00	1.00	0.91	0.78	0.75	0.83	0.43	0.71	0.77	0.83
	1.00	1.00	0.91	0.78	0.75	0.83	0.43	0.71	0.77	0.83
min	*0.71*	*0.43*	*0.44*	*0.33*	*0.19*	*0.13*	*0.21*	*0.07*	*0.39*	*0.24*
max	*1.00*	*1.00*	*0.91*	*0.78*	*0.75*	*0.83*	*0.43*	*0.71*	*0.77*	*0.83*
average	*0.90*	*0.78*	*0.61*	*0.59*	*0.55*	*0.56*	*0.32*	*0.44*	*0.59*	*0.59*
std	*0.12*	*0.20*	*0.16*	*0.16*	*0.18*	*0.23*	*0.09*	*0.23*	*0.13*	*0.20*
p T<t	*0.07*		*0.66*		*0.83*		*0.06*		*1.00*	

Partitioned by Exercise

	Appropriate patterns		Potential patterns		Appropriate links		Potential links		Overall	
	ex1	ex2	ex1	ex2	ex1	ex2	ex1	ex2	ex1	ex2
	0.43	0.77	0.33	0.45	0.42	0.13	0.29	0.07	0.37	0.36
	0.43	0.77	0.33	0.45	0.42	0.13	0.29	0.07	0.37	0.36
	0.64	0.78	0.44	0.45	0.50	0.19	0.43	0.21	0.50	0.41
	0.64	0.78	0.44	0.45	0.50	0.19	0.43	0.21	0.50	0.41
	0.67	1.00	0.44	0.64	0.50	0.57	0.43	0.21	0.51	0.61
	0.71	1.00	0.56	0.64	0.50	0.57	0.43	0.21	0.55	0.61
	0.71	1.00	0.56	0.64	0.50	0.60	0.43	0.21	0.55	0.61
	0.80	1.00	0.56	0.64	0.50	0.60	0.43	0.21	0.57	0.61
	0.80	1.00	0.56	0.73	0.60	0.60	0.43	0.29	0.60	0.65
	0.83	1.00	0.56	0.73	0.75	0.64	0.57	0.29	0.68	0.66
	0.86	1.00	0.67	0.73	0.75	0.71	0.71	0.36	0.75	0.70
	0.86	1.00	0.67	0.73	0.80	0.71	0.71	0.36	0.76	0.70
	1.00	1.00	0.78	0.91	0.83	0.75	0.71	0.43	0.83	0.77
	1.00	1.00	0.78	0.91	0.83	0.75	0.71	0.50	0.83	0.79
min	*0.43*	*0.77*	*0.33*	*0.45*	*0.42*	*0.13*	*0.29*	*0.07*	*0.37*	*0.36*
max	*1.00*	*1.00*	*0.78*	*0.91*	*0.83*	*0.75*	*0.71*	*0.50*	*0.83*	*0.79*
average	*0.74*	*0.94*	*0.55*	*0.65*	*0.60*	*0.51*	*0.50*	*0.26*	*0.60*	*0.59*
std	*0.18*	*0.11*	*0.14*	*0.15*	*0.16*	*0.24*	*0.16*	*0.12*	*0.15*	*0.15*
p T<t	*0.002*		*0.081*		*0.249*		*0.000*		*0.881*	

A16.2 Timing

To Complete Exercises

	Exercise 1	Exercise 2	Average
Illustrated	28	24	26
Narrative	32	23	28
Average	30	24	

To Complete UI-pattern Models

	Exercise 1	Exercise 2	Average
Illustrated	25	24	24
Narrative	32	23	28
Average	28	23	

A16.3 Patterns Questionnaire

Correlation Table

	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A
1B	0.**386**												
2A	0.300	0.222											
2B	0.291	0.227	0.**381**										
3A	0.068	-0.084	0.080	0.092									
3B	0.225	-0.025	0.117	0.251	0.**275**								
4A	-0.113	-0.203	0.052	0.235	0.447	0.361							
4B	0.062	0.038	0.138	0.159	0.347	0.304	0.**547**						
5A	0.007	-0.081	0.208	0.224	0.289	0.332	0.470	0.411					
5B	0.200	0.150	0.040	0.318	0.270	0.238	0.329	0.337	0.**186**				
6A	0.101	-0.050	0.351	0.337	0.225	0.197	0.348	0.405	0.319	0.412			
6B	0.179	-0.006	0.000	0.267	0.166	0.291	0.232	0.284	0.331	0.414	0.**398**		
7A	0.130	-0.024	0.318	0.386	0.159	0.165	0.302	0.295	0.105	0.335	0.321	0.459	
7B	0.063	-0.088	0.144	0.206	0.349	0.216	0.540	0.466	0.214	0.272	0.304	0.256	0.**257**

Patterns Questionnaire Raw Data

Num	Topic	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A	7B	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A	7B
		Illustrated Patterns														Narrative Patterns													
		Exercise 1						Exercise 2								Exercise 2						Exercise 1							
S-01	Clear information	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	-1
S-02	Patterns helpful	0	1	1	1	-1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	0	0	0
S-03	Straight forward to select pattern	-1	-1	-1	-1	-1	-1	-1	-1	1	0	-1	1	0	0	-1	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1
S-04	Enjoyed building UI pattern model	0	0	0	0	0	1	1	1	1	-1	0	0	0	0	0	-1	1	0	0	0	-1	1	1	-1	0	1	0	0
S-05	Finding suitable patterns OK	-1	0	-1	-1	0	0	1	1	0	0	-1	1	0	1	-1	0	0	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
S-06	Find required knowledge	0	1	-1	-1	-1	-1	-1	0	-1	0	-1	0	0	-1	0	-1	0	0	-1	-1	0	-1	0	-1	-1	0	-1	-1
S-07	Patterns informative	0	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	1	1	1	0	1	1	0	0	0	1	1	0
S-08	Patterns mentally stimulating	-1	-1	-1	0	1	-1	1	1	0	1	0	1	1	1	-1	-1	0	0	1	-1	1	1	0	1	1	0	0	1
S-09	Sufficient information	1	1	1	1	0	1	1	1	1	1	0	1	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	1
S-10	Easy to find next pattern	0	1	-1	0	0	1	1	0	1	1	-1	1	0	0	0	1	0	0	1	0	-1	0	0	0	-1	-1	-1	0
S-11	Command of patterns	0	-1	-1	0	1	1	1	1	1	0	0	0	0	0	0	1	1	-1	1	-1	-1	-1	0	-1	0	0	0	0
S-13	Liked to use patterns designing UI	-1	-1	0	1	0	-1	1	0	1	0	0	1	0	-1	-1	1	1	1	-1	0	-1	0	1	0	1	1	1	-1
S-14	Straight forward method	1	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	-1	-1	1	1	1	0	0	0	-1
S-15	Aid UI-concepts discussion	-1	-1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	0	1	0	1	1	1	1	1	1	1	0
S-17	Referring back to patterns	-1	0	1	1	-1	-1	1	1	1	0	0	-1	1	0	0	1	1	0	0	-1	1	1	1	0	1	1	0	1
S-18	Patterns easy to understand	0	1	0	0	0	0	1	1	1	1	1	1	0	0	0	1	1	0	-1	1	-1	0	0	1	1	1	0	-1
S-19	Confident about UI knowledge	-1	-1	-1	-1	1	0	1	0	1	1	1	1	0	0	-1	-1	1	-1	0	-1	1	1	1	1	1	0	0	0
S-20	Patterns described UI well	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0	1	0	0	1	-1	0	0	1	0	-1	1	1	-1
S-21	Logical organisation	0	1	0	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	0	-1	-1	-1	0	1	-1	1	1	0
S-22	Patterns easy to use	-1	1	-1	0	0	-1	1	1	1	1	0	1	0	-1	0	1	0	0	-1	-1	-1	0	0	1	1	1	0	-1
S-23	Number of steps OK	0	1	0	0	0	-1	1	0	1	0	0	0	-1	0	0	1	0	0	0	-1	-1	0	1	0	0	0	-1	0
S-24	Enough patterns to model UI	-1	1	0	0	1	1	1	1	1	1	-1	0	0	-1	-1	1	1	0	1	-1	1	1	1	0	1	0	0	0
S-25	Content easy to remember	0	1	-1	1	0	-1	-1	1	-1	1	1	1	0	0	1	0	0	1	0	0	0	-1	0	1	0	0	0	-1
S-27	Aid modelling discussion	0	0	0	1	1	0	1	0	1	1	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	0	1
S-28	Method easy to follow	1	1	0	0	0	0	-1	1	1	1	0	1	0	-1	1	1	0	-1	0	-1	0	0	0	0	0	0	0	-1
S-30	Patterns aid focus	0	-1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1	-1	1	0	1	1	0	1	0	1	1	0
S-01	Clear information	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0	1	1	1	-1
S-02	Patterns helpful	0	1	1	1	-1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	0	0	0

Patterns Questionnaire Summary Data

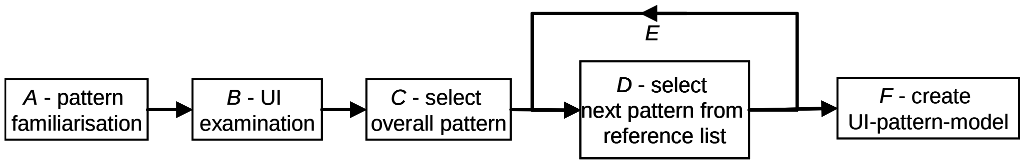
Num	Topic	Total	agree	undecided	disagree
S-01	Clear information	17	64%	32%	4%
S-02	Patterns helpful	18	68%	29%	4%
S-03	Straight forward to select pattern	-18	7%	21%	71%
S-04	Enjoyed building UI pattern model	4	29%	57%	14%
S-05	Finding patterns OK	-11	14%	32%	54%
S-06	Find required knowledge	-15	4%	39%	57%
S-07	Patterns informative	18	64%	36%	0%
S-08	Patterns mentally stimulating	6	46%	29%	25%
S-09	Sufficient information	20	71%	29%	0%
S-10	Easy to find next pattern	2	29%	50%	21%
S-11	Command of patterns	1	29%	46%	25%
S-13	Liked to use patterns designing UI	3	39%	32%	29%
S-14	Straight forward method	12	54%	36%	11%
S-15	Aid UI-concepts discussion	17	68%	25%	7%
S-17	Referring back to patterns	9	50%	32%	18%
S-18	Patterns easy to understand	10	46%	43%	11%
S-19	Confident about UI knowledge	4	43%	29%	29%
S-20	Patterns described UI well	9	43%	46%	11%
S-21	Logical organisation	12	57%	29%	14%
S-22	Patterns easy to use	2	36%	36%	29%
S-23	Number of steps OK	0	18%	64%	18%
S-24	Enough patterns to model UI	9	50%	32%	18%
S-25	Content easy to remember	3	32%	46%	21%
S-27	Aid modelling discussion	18	64%	36%	0%
S-28	Method easy to follow	3	29%	54%	18%
S-30	Patterns aid focus	14	57%	36%	7%

A16.4 Exit Questionnaire – Closed Questions

	PROFICENCY		**COMMUNICATION AID**	**UI-PATTERN METHOD**	**DESIGN AID**
Experienced	1	Very helpful	2	6	2
Reasonable	3	Reasonably helpful	6	2	4
Average or OK	6	Helpful	4	6	6
Not very	2	Not very helpful	2	0	1
Low	2	Unhelpful	0	0	1
		min	0	0	1
		max	6	6	6
		median	2	2	2
		mean	2.80	2.80	2.80
		Std	2.28	3.03	2.17
		skew	0.40	0.32	0.91
		kurtosis	-0.18	-3.08	-0.74

Appendix A17: Study One – UI-pattern Modelling Case Studies

Three pairs of students used the illustrated pattern form first and are identified as cases 1, 2 and 3. Four pairs of students used the narrative pattern form first and are identified as cases 4, 5, 6 and 7. Demographic data for students in each case is not provided because the sample size is so small that individuals may be identifiable from such a description.



A. Become familiar with the subset of UI patterns.
B. Examine the illustration of the user interface for the exercise.
C. Select a pattern from the first three patterns in the subset (1, 2 or 3) that best describes the overall nature of the type of data or information that the UI deals with.
D. From the list of patterns that this pattern references, select those that best describe some other feature of the user interface you have been asked to model.
E. Continue in a similar manner selecting more patterns from the list of patterns referred to at the end of each pattern you select; select patterns that best describe each feature of the user interface until you can't select any more
F. Using the patterns you have selected create a diagram showing those patterns and connecting them with lines to indicate how they are linked into a UI-pattern model.

Figure A17.1 – The UI-pattern modelling method used for Study One

The data for each case is described using the same pattern . First there is a description of the artefacts the students are using at the time the image is taken accompanied by an interpretation of what they are appeared to be doing. Then the students use of the illustrated pattern set is compared with their use of the narrative pattern language. The discussion concludes with an interpretation of the method the students appear to have followed and this is compared with the given UI-pattern modelling method. Figure A17.1 shows the method with the details for each step.

Each of the seven cases is described in turn. For each case, exercise one is described first followed by exercise two.

A17.1 Case 1 - Illustrated patterns First

	5 mins		10 mins	15 mins	18 mins	22 mins	24 mins
ILLUSTRATED	Pattern list UI Instructions Exercise Sheet Pile of patterns *Pointing to illustration*		UI Pile of patterns Original collection Focus on a pattern within collection	UI Instructions Pile of patterns Original collection	UI partly hidden Instructions Selected patterns laid out overlapping Researcher pointing to reference section	Instruction UI partly hidden Selected patterns laid out overlapping Partial graph	Finished
	5 mins	8 mins	12 mins	15 mins	18 mins	20 mins	23m
NARRATIVE-ONLY	Pattern list UI Instructions Exercise Sheet Original collection Pile of patterns *Pointing to component on UI*	Pattern list UI Exercise Sheet Pile of patterns *Pointing to the reference section*	Pattern list UI Exercise Sheet Pile of patterns	UI Pile of patterns *Pointing to the solution section*	UI Exercise Sheet Pile of patterns Selected patterns laid out as graph	UI Pattern list Pile of patterns Selected patterns laid out as graph Partial graph	Finished

Table A17.1- Comparison of observations for Case 1

The description of the items being pointed, to which may indicate which sections of the patterns the students have found most useful, will be left to the concluding discussion. When working with the illustrated patterns, after the first five minutes the students have laid out the artefacts listed in Table A17.1. It is not clear how long the students spent examining the sets of patterns if at all before they moved focus to the exercise UI. They worked for fifteen minutes examining the UI and selecting patterns, apparently matching the initial illustration at the beginning of each patterns to UI components to guide their selection process. Eighteen minutes into the exercise this pair requested an explanation about how to link the patterns they had selected. Within four minutes of obtaining help they have selected patterns spread out on the work surface and are drawing their UI-pattern-model diagrams.

When working with the narrative patterns it appears that the students are initially following a similar method to that used for the illustrated patterns, although more pointing behaviour was captured in the images. By fifteen minutes the students appear to be organising their selected patterns and laying them out as a hierarchy representing the UI-pattern-model. They then draw the model on their exercise worksheets.

Comparing behaviour working on the two exercises the students appear to initially focus on the UI because it is in a dominant position after five minutes. In both exercises the students identified a pattern that describes one of the more prominent UI components rather than using the proposed top-down approach. Examining the names of the selected patterns in each image indicates that the UI components were an important guide at the early stages.

When working with the illustrative set it is not until they have exhausted their selections that they try to find the links between the selected patterns. Pointing behaviour when working with the narrative patterns shows that they used link information in the context and reference sections earlier.

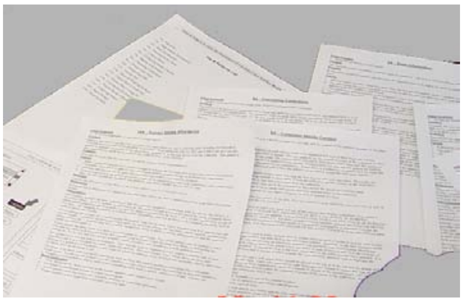


Figure A17.2 - Building a pattern hierarchy

When working with the narrative patterns, they constructed the UI-pattern-model by manipulating the physical patterns, laying them out in a hierarchy before they began drawing on their exercise worksheets as shown in Figure A17.2. The other obvious difference between how they worked on the two exercises is that the examination of pattern content appears to have been postponed when working with the illustrated patterns.

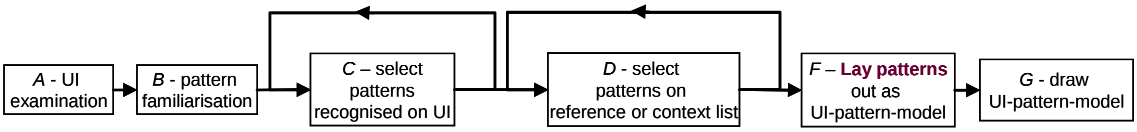


Figure A17.3 - Apparent method used by pair 1

The method the students appear to have used for the second exercise is shown in Figure A17.3. They appeared to reverse the first two steps (A and B) of the proposed method and examined the UI before becoming familiar with the patterns. Then rather than use a top-down approach they have selected patterns that matched UI components that they recognised on the exercise UI, step C. Using this set of patterns they then appear to have then used the context and reference sections to select more patterns. They are seen reviewing the patterns as they lay the patterns out in a hierarchical structure, steps E and F. Finally they make a copy of their model on the exercise sheet.

A17.2 Case 2 - Illustrated patterns First

	5 mins		10 mins	14 mins	18 mins	21 mins	25 mins	28 mins	30m
ILLUSTRATED	Pattern List Original collection Focus on a pattern within collection *Pointing to the reference section*		Selected patterns laid out over-lapping	Instructions UI Selected patterns laid out over-lapping	UI Pattern list Selected patterns laid out over-lapping Rough graph	Pattern list Exercise worksheet Selected patterns laid out overlapping Rough graph Partial graph	Pattern list Original collection Selected patterns laid out over-lapping Completed rough graph Partial graph	UI Pattern list Pile of patterns Selected patterns laid out over-lapping Partial graph	Finished
	5 mins	8 mins	12 mins	15 mins	18 mins	20 mins	23 mins		
NARRATIVE-ONLY	UI Pile of patterns Instructions Focus on a pattern *Pointing to forces section*	UI Pile of patterns	UI Instructions Pile of patterns *Pointing to the UI*	UI Pile of patterns *Pointing to the reference section*	UI Selected of patterns laid out over-lapping Partial graph	UI Instructions Selected patterns laid out over-lapping Partial graph	Finished		

Table A17.2- Comparison of observations for Case 2

The lists of artefacts identified in the images taken throughout the exercises as well as the students' point of focus are shown in Table A17.2. This student pair appears to start by studying the collection of illustrated patterns. They are first seen studying the high level pattern 'State Information' and one student is pointing to the reference section.

They do not appear to consider the UI until fourteen minutes into the exercise and by eighteen minutes, as can be seen in Figure A17.4, both students are working on a rough copy of the UI-pattern-model recording just the pattern id numbers.

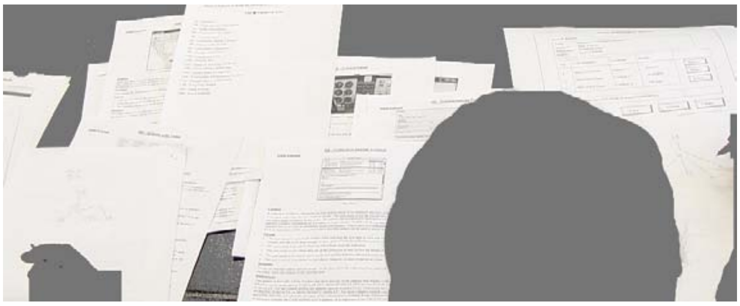


Figure A17.4 - Working on rough copies of the UI-pattern-model

They have completed the rough copies by twenty-one minutes. Then they appear to spend most of the last ten minutes reviewing their selected patterns and copying the rough copy onto the exercise sheet.

When working with the narrative patterns in the first image the students appear to be comparing the content of two patterns then at eight minutes the UI is clearly visible beside the patterns they have selected. They are drawing the UI-pattern model graph by eighteen minutes and one student appears to moving the patterns, once recorded, from the central pile across to the left.

The most obvious difference when comparing work with the two pattern sets is the time. This pair took longer to complete the first exercise. However if the students had worked directly on the exercise sheet during the first exercise, as they did for the second exercise, and not spent so much time reviewing their work, they may have completed the two exercises in a similar time frame. All the same, the students appear to have spent relatively significant time in both exercises considering pattern content and also discussing the patterns they have selected. In both exercises they clearly reviewed the patterns they had selected as they drew the final UI-pattern-model.

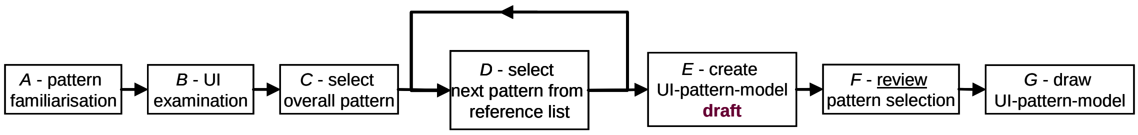


Figure A17.5 - Apparent method used by Case 2

The students appear to have followed the suggested top-down approach with both pattern sets and this supposition is supported by notes showing that they asked for clarification about which patterns they should select the first pattern from. They followed the proposed method for the first four steps, and as Figure A17.5 shows they inserted two additional steps. They completed a rough copy UI-pattern-model, step E then reviewed their work, step F, and finally redrew the model on the worksheet, step G.

A17.3 Case 3 - Illustrated patterns First

	6 mins		10 mins	14 mins	18 mins	22 mins	25 mins	28 mins	31 mins
ILLUSTRATED	Instructions Original collection *Pointing to instructions*		Pattern list UI Instructions Original collection Pile of patterns	UI Pile of patterns Selected patterns laid out overlapping *Pointing to component on UI* *Pointing to context section*	Pattern list UI Selected patterns laid out overlapping	Pattern list covered UI Selected patterns laid out overlapping Partial graph	Pattern list covered UI Selected patterns laid out overlapping Completed graph	UI Selected patterns laid out overlapping Completed pattern graph *Pointing to the reference section (on two patterns)*	Finished
	5 mins	8 mins	12 mins	15 mins	18 mins	20 mins	23 mins		
NARRATIVE-ONLY	Pattern list UI Original collection *Pointing to the forces section*	UI Pile of patterns Exercise Sheet *Pointing to description section*	UI Selected patterns laid out overlapping Instructions Exercise Sheet	Pattern list UI Selected patterns laid out overlapping Partial graph *Pointing to reference section*	Selected patterns laid out overlapping Partial graph *Pointing to discussion section*	Pattern list UI Instructions Partial graph Selected patterns laid out overlapping	Finished		

Table A17.3 - Comparison of observations for Case 3

The lists of artefacts identified in the series of images for this pair of student are shown in Table A17.3. After the first five minutes the original collections of illustrated patterns are still fixed in order by the paper clip and both students are studying the instructions. These students then spend over fifteen minutes examining the patterns and the UI.

At twenty-five minutes the students have completed the graphs, as handed in and appear to move onto a review phase which can be seen in Figure A 17.6 where they have two patterns side by side. They then spent six minutes reviewing their work.

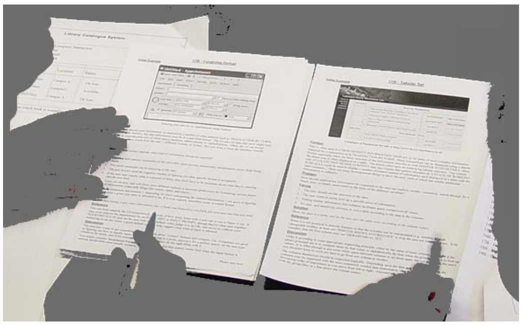


Figure A 17.6 - Comparing patterns

When working with the narrative patterns the images of these students capture a lot of pointing behaviour and it is clear that much discussion of pattern content was occurring. For example, an image that shows one student holding the pattern and pointing to the reference section. At eighteen minutes the students have started drawing a graph and this is the only image in which the UI is not visible. They finish the exercise within twenty-three minutes.

The time taken to complete the exercise is the most obvious difference when working with the two types of pattern. They spent significantly more time reviewing their UI-pattern-models during the first exercise.

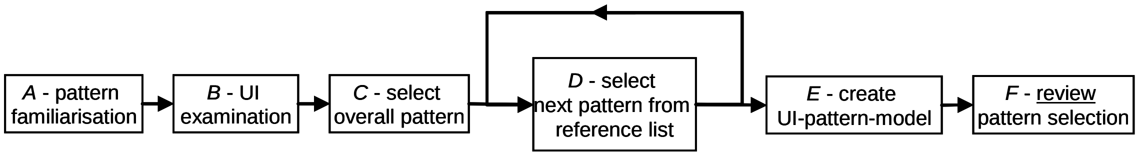


Figure A17.7 - Apparent method used by Case 3

The students appear to have followed the top-down approach recommended in the proposed method but added a review step at the end as shown in Figure A17.7.

A17.4 Case 4 – Narrative patterns First

	6 mins		**11 mins**	**15 mins**	**18 mins**	**22 mins**	**25 mins**	**28 mins**	**31 mins**	**33 mins**
NARRATIVE-ONLY	Original collection Exercise Sheet		Pattern list UI Instructions Selected patterns laid out overlapping Exercise Sheet *Pointing to component on UI*	Pattern list UI Pile of patterns Exercise worksheet *Pointing to the problem section*	UI Selected patterns laid out overlapping Rough graph	UI Pattern list Selected patterns laid out overlapping Rough graph	UI Pattern list Selected patterns laid out overlapping Rough graph	UI Selected patterns laid out overlapping Rough graph *Pointing to the forces section*	Patterns on list marked UI Instructions Selected patterns laid out overlapping Partial graph *Pointing to component on UI*	Finished
	5 mins	**8 mins**	**12 mins**	**15 mins**	**18 mins**	**21 mins**	**24 mins**			
ILLUSTRATED	UI Original collection List *Pointing to component on UI*	Pattern list UI Pile of patterns	Patterns on list marked UI Pile of patterns *Pointing to component on UI*	Patterns on list marked UI partly hidden Pile of patterns Rough graph *Pointing to pattern name on list*	Patterns on list marked Pile of patterns Rough graph	Patterns on list marked Pile of patterns Rough graph *Pointing to pattern name on list*	Finished			

Table A17.4 - Comparison of observations for Case 4

The lists of artefacts identified in the images taken throughout the exercises as well as the students' pointing behaviour are listed in Table A17.4. This pair when working with the narrative patterns, start by becoming familiar with them and don't appear to focussing on the UI until about eleven minutes have elapsed. They have begun a rough copy of the graph by fifteen minutes. Next the students are observed studying two patterns that are recorded on their rough graph and it is assumed that they are trying to determine how the patterns are connected but on the next image the rough graph has many edges drawn on it with a large cross though it. Notes show that at this point the students requested help in how to link patterns. They are finished by thirty-three minutes.

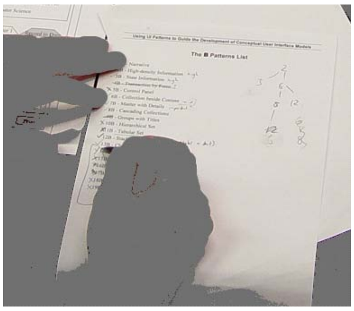


Figure A17.8 - Working directly on pattern list

When working with the illustrated patterns the students appear to predominantly focus on the UI and the illustrations. They also used the technique of recording patterns as selected or rejected on the pattern list. They had successfully used this approach with the narrative patterns. At fifteen minutes the students have marked patterns on the list as selected or rejected and are creating a rough copy of UI-pattern-model to the left of the list as shown in Figure A17.8.

By twenty-one minutes it appears that the students are reviewing their choices and have finished within twenty-four minutes

These students took nine minutes longer to complete the first exercise. During that exercise they appear to have had difficulty completing the exercise until they sought help. At that point they changed their approach to trying to link the patterns and began using the pattern list as a guide to choices. They then used this approach again using different forms of marking on the list as they developed the UI-pattern-model. They seem to rely on the illustrations during the second exercise rather than pattern content.

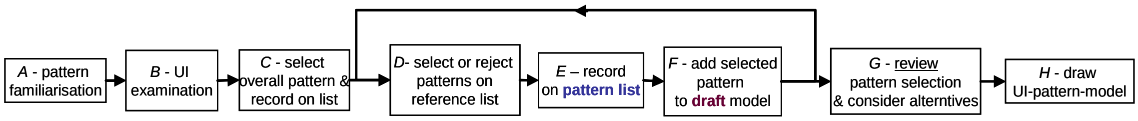


Figure A17.9 - Apparent method used by Case 4

Figure A17.9 shows the modification these students have made to the proposed method. Steps A and B follow the proposed method but step C has been extended to include marking the selected pattern on the pattern list. Steps D and E work together as the students make decisions for and against patterns they are marked on the pattern list including adding comments. Once patterns are selected they are added to the draft UI-pattern-model. Next at step G, the patterns used for the draft model are reviewed and alternatives considered. Finally a good copy of the model is drawn.

A17.5 Case 5 – Narrative patterns First

	5 mins		**11 mins**	**15 mins**	**19 mins**	**22 mins**	**25 mins**
NARRATIVE-ONLY	UI Instructions Exercise sheet Selected patterns laid out overlapping Partial graph		UI Instructions Exercise sheet Selected patterns laid out overlapping *Pointing to the reference section*	UI Pattern list Exercise sheet Pile of patterns *Pointing to component on UI*	UI Pattern list Instructions Pile of patterns Partial graph	UI Patterns on list marked Pile of patterns Partial graph	Finished
	6 mins	**9 mins**	**13 mins**	**16 mins**	**19 mins**	**22 mins**	
ILLUSTRATED	UI partly hidden Instructions Selected patterns laid out overlapping Partial graph	Pattern list UI Instructions Selected patterns laid out overlapping Partial graph *Pointing to component on UI*	UI Pile of patterns Partial graph	UI Instructions Pile of patterns Partial graph	UI Instructions Pattern list Pile of patterns Partial graph Completed graph	Finished	

Table A17.5 - Comparison of observations for Case 5

Table A17.5 details the lists of artefacts identified in the images taken throughout the exercises as well as the students' pointing behaviour. The students appear to have made a false start with creating the UI-pattern-model and then spent the next fourteen minutes of this exercise studying the narrative patterns. They begin drawing a rough UI-pattern-model at nineteen minutes but have the pattern list central on their work surface. The rough graph clearly shows links between the selected patterns that are referred to in the context section of their selected patterns. The students are working on the final copies of their UI-pattern-models at twenty-two minutes and have completed them within twenty-five minutes. Interestingly, the final graphs they are seen working on are not the same as the rough copy. There may have been a review process but there is no other evidence that this took place.

The students wasted no time getting started using the illustrated patterns. By five minutes they have selected the top level 'State Information' pattern as the root of a rough graph and by nine minutes there is a list of pattern numbers beside that rough graph which now contains nine patterns. By nineteen minutes both students have their own copies of the UI-pattern-model and one student has completed. The other is finished by 22 minutes.

The main difference in the students' use of the two types of pattern seems to be in the speed with which they started creating a rough copy UI-pattern-model when working with the illustrated patterns.

Although there is not clear evidence there is the suggestion within the images that in both cases one student marked the selected patterns off on the patterns list as the patterns are selected. When working with the illustrated patterns the other student created the list by recoding the pattern id's alongside the rough copy as shown in Figure A17.10.

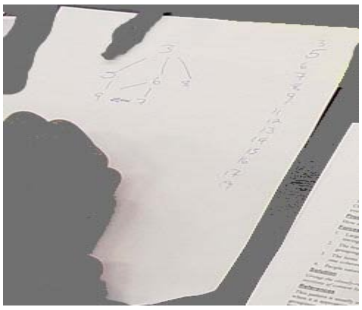


Figure A17.10 – Rough copy beside list of selected patterns

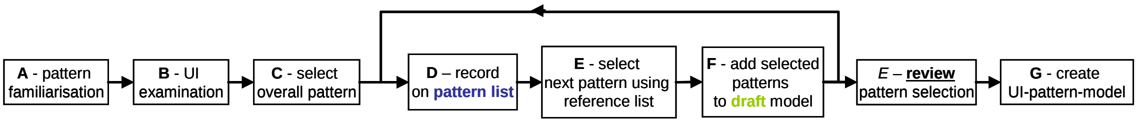


Figure A17.11 - Apparent method used by Case 5

The images showing the development of the rough copies indicate that in both exercises they have mostly followed the proposed top-down method with one variation. As can be seen in Figure A17.11 they have inserted step D where they record the patterns they select on the pattern list as well as keeping them in a separate pile.

A17.6 Case 6 – Narrative patterns First

	6 mins	11 mins		15 mins	19 mins	23 mins	25 mins	29 mins	31 mins	33
NARRATIVE-ONLY	UI Instructions Pile of patterns Exercise Sheet *Pointing to the solution section*	UI Pattern list Pile of patterns Partial graph		UI Pattern list Selected patterns laid out overlapping Partial graph *Pointing to the reference section*	UI Pattern list Selected patterns laid out overlapping *Pointing to the reference section*	UI Pattern list Instructions Selected patterns laid out overlapping Partial graph *Pointing to the solution section*	UI Patterns on list marked Instructions Selected patterns laid out overlapping Partial graph	UI Partial graph Selected patterns laid out as graph	UI Patterns on list marked Selected patterns laid out as graph Partial graph	Finished
	6 mins	10 mins	13 mins	16 mins	19 mins	22 mins	25 mins			
ILLUSTRATED	Pattern list UI Instructions Pile of patterns Selected patterns laid out overlapping	Pattern list UI Instructions Pile of patterns *Pointing to the description section*	Patterns on list marked UI Instructions Pile of patterns *Pointing to the solution section*	Patterns on list marked UI Instructions Pile of patterns *Pointing to the discussion section*	Pattern on list marked UI Instructions Pile of patterns	Patterns on list marked UI Instructions Selected patterns laid out as graph Partial graph	Finished			

Table A17.6 - Comparison of observations for Case 6

The artefacts students have laid out on the work surface and pointing behaviour when each image was taken are listed in Table A17.6.

When working with the narrative patterns these students both started on graphs and then stopped work on them sometime between six and eleven minutes and although the graphs are observed in a number of subsequent images they effectively spent twenty-nine minutes studying the patterns before returning to work on their graphs.

The series of images between six and twenty-five minutes show different subsets of patterns spread over the worktop then the layout changes into a hierarchical structure at twenty-nine minutes. Figure A17.12 shows the students adding a third level of patterns to the hierarchy at thirty-one minutes. They have completed the exercise by thirty-three minutes.

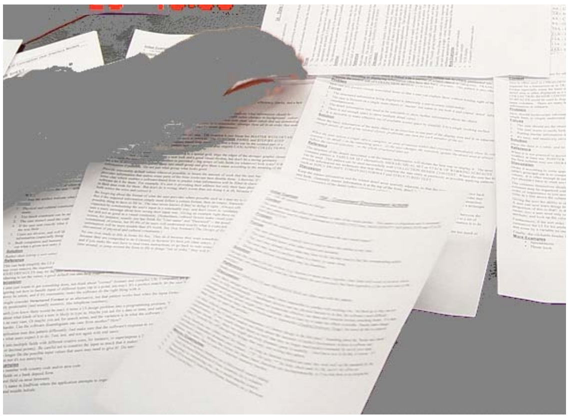


Figure A17.12 - Adding next level to hierarchy of patterns

In the second exercise, after six minutes the students are examining the top two patterns in the set of illustrated patterns with the pattern list and UI clearly visible. By thirteen minutes selected patterns have been marked on the pattern list repeating the technique seen first in the last six minutes of Exercise One. A change in the work surface is seen at nineteen minutes with students apparently using the marked pattern list to locate the patterns and organise them into a graph like they did with the narrative patterns. This process is almost complete at twenty-two minutes and the students are working on their UI-pattern models which have been completed by twenty-five minutes.

When comparing the process the students used for the two exercises it is obvious that the way they manipulated the patterns was similar in the last five minutes where they effectively moved the patterns into a hierarchical structure. The main difference is in the time they took when working with the narrative patterns to examine and select patterns and also the apparent false start on the graph. The other obvious difference is that they marked their selected patterns on the list when working with the illustrated patterns before building the hierarchy. Because the students are not observed examining the UI, and it does not appear to have been moved into the central area at any time, it is assumed that they examined this in detail during the first time period and only made reference to it as they selected patterns to describe it.

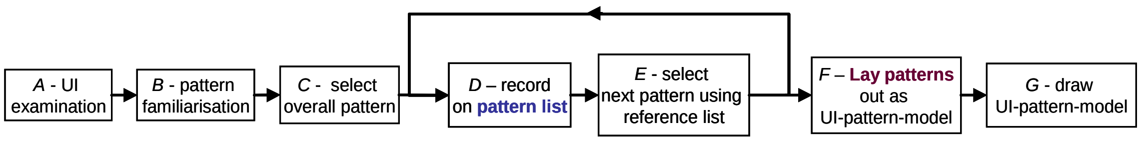


Figure A17.13 - Apparent method used by Case 6

The method these students appear to have created, Figure A17.13, interchanges steps A and B when compared to the proposed method. They then used the suggested top-down approach but these students marked selected patterns on the pattern list as patterns are selected, steps D and E. Then using that list they have reorganised the patterns into a hierarchy. Finally they record the UI-pattern model on exercise worksheet.

A17.7 Case 7 – Narrative patterns First

	7 mins	**11 mins**	**15 mins**		**19 mins**	**23 mins**	**26 mins**	**29 mins**	**31 mins**	**35**
NARRATIVE-ONLY	UI Instructions Selected patterns laid out overlapping Exercise Sheet	UI Instructions Selected patterns laid out overlapping Exercise Sheets	UI Instructions Selected patterns laid out overlapping Exercise Sheet		UI Instructions Selected patterns laid out overlapping Partial graph *Pointing to the reference section*	Instructions Selected patterns laid out overlapping Partial graph	UI Instructions Patterns on list marked Selected patterns laid out as graph Partial graph *Pointing to context section*	UI Instructions Patterns on list marked Selected patterns laid out as graph	Instructions Patterns on list marked Selected patterns laid out as graph	Finished
	7 mins	**10 mins**	**14 mins**	**17 mins**	**20 mins**	**23 mins**	**26 mins**			
ILLUSTRATED	Pattern list UI Instructions Pile of patterns Selected patterns laid out overlapping	UI Pile of patterns Selected patterns laid out overlapping *Pointing to the reference section*	UI Pile of patterns Selected patterns laid out overlapping *Pointing to the illustration*	Instructions Pattern list UI Pile of patterns Selected patterns laid out overlapping	Instructions Pattern list UI Pile of patterns Selected patterns laid out overlapping Partial graph	Instructions Pattern list UI Pile of patterns Selected patterns as graph Partial graph Completed rough graph	Finished			

Table A17.7 - Comparison of observation for Case 7

The list of artefacts and items of interest for this pair of students are found in Table A17.7. As this group was the last pair in the circuit for photographing seven minutes elapsed before the first image was taken. By this stage the students had the first few patterns from the pattern set laid out in order across the work surface and one student appears to be studying the method on the instruction sheet. Notes show that they had asked for an explanation of the instructions earlier. At eleven minutes they requested help again and asked about the different sections making up the patterns. At fifteen minutes one student had started to create the graph but at nineteen minutes it is clear the graph has not changed. The students appear to re-organising selected patterns for about ten minutes and notes show that they again requested help. At about twenty-three minutes they asked how the patterns could be structured into the UI-pattern-model and as they had observed the group behind them laying out the patterns into a hierarchy wanted to know if this approach would work. One of the students also asked about the inconsistency between the context and reference sections of patterns ‘09 Groups with Titles’ and ’07 Master with Details’. At twenty-six minutes the students are marking selected patterns on the list, have developed three layers of the pattern hierarchy and are drawing the matching UI-pattern-model. They appear to have extensively discussed pattern content as they take nine more minutes to complete the exercise.

When using the illustrated patterns this pair of students seems to start selecting patterns using the UI and the pattern list as seen in the first image. At ten minutes the UI is still prominent and they appear to be comparing two patterns. In Figure A17.14 they are seen comparing the UI with the pattern's illustration.

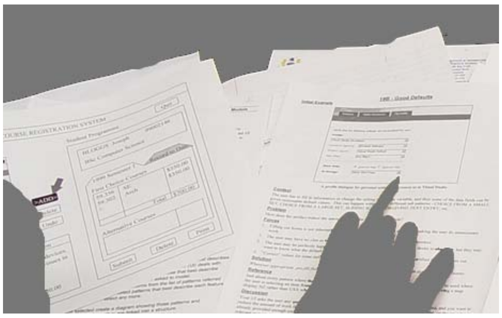


Figure A17.14 - UI being compared with pattern illustration

Comparing patterns seems to be the main behaviour observed in the series of images. It is not until twenty-three minutes that it can be seen that one student has drawn a four layer UI-pattern-model. The other student has drawn a rough copy over the top of the instruction sheet and has just started transferring that to the worksheet. Both have finished the exercise by twenty-six minutes.

When working on the first exercise, this pair of students requested help on a number of occasions. They were much quicker getting started and completing the second exercise. When working with both pattern sets the students made frequent reference to the UI throughout the exercise. They made use of the pattern list only in the final stages of both exercises. When working with the narrative patterns these students spent considerable time discussing possible alternatives as they made selections. They do not seem to have had similar discussions while completing the second exercise though it is clear one student created a rough copy while the second worked directly on the final version of their graph.

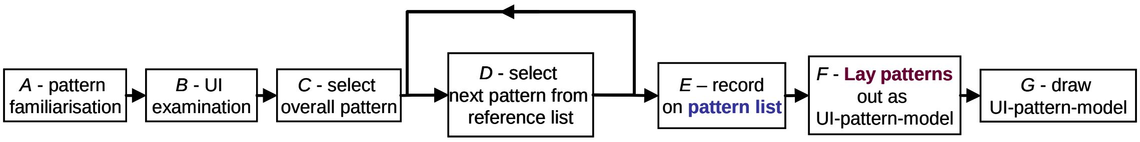


Figure A17.15 - Apparent method used by Case 7

The method these students appear to use when working with the narrative patterns is shown in Figure A17.15. For the first four steps they appear to follow the proposed method. They may have used the UI to identify patterns rather than the reference section in patterns as the UI was obviously being referred to a lot but there is no direct evidence they were doing this. They appear to spend a significant amount of time comparing patterns before they marked the selected patterns on the list but not all patterns used in their graphs are identified. This maybe a result of the students not agreeing to which patterns make up the lower level of their graphs as their completed solutions are very different.

Appendix A18: Study One – Exit Questionnaire Responses

To ensure anonymity the comments have been amalgamated and sorted alphabetically.

A18.1 UI-pattern Patterns

Justification for most useful pattern section

⇒ Because once we had identified the patterns present we used these attributes to relate them (context & reference)
⇒ Because the problem identifies what it is solving in the context
⇒ Examples gave me an idea of what they would look like
⇒ Facilitated understanding (Initial example & context)
⇒ It displayed the correct solution which could be visualised to help me figure out if that same solution could be applied to exercise we were given (Solution)
⇒ Most relevant to task (Initial example & context)
⇒ Provided more detailed explanation (Problem & solution)
⇒ Reference because it shows relationships
⇒ Showed how the patterns linked together (reference)
⇒ Showed how they linked together (Reference)
⇒ So you could see what that pattern looked like (Initial example & more examples)
⇒ Solutions show how they are used
⇒ The context and examples gave overviews of the differences between patterns
⇒ The forces provided links between patterns
⇒ The wording instructions were too complicated (Initial example & more examples)
⇒ They gave a feel for how you'd stack them in the hierarchy. Also helped match to the parts of the UI (context & reference)
⇒ Undecided

Justification for least useful pattern section

⇒ Not relevant, in most cases it was clear from the name and initial example (Forces)
⇒ Given the time, the discussion would be better for non-group works. We just used the rest of the description to match and provided our own commentary.
⇒ Ran out of time to read them (Discussion)
⇒ Lead to reliance on them – used them to understand the pattern not the information explaining them (Initial example & more examples)
⇒ Didn't focus on them (more examples)
⇒ Because there was insufficient time to analyse the discussion
⇒ Not used redundant (forces, discussion, identifier & name)
⇒ Didn't use it (forces)
⇒ Too much useless information (Discussion)
⇒ It was long and I couldn't be bothered reading it (Discussion)

Illustrated Patterns Preference

⇒ Examples
⇒ No screen shots
⇒ No screen shots in the second set of patterns made the non-illustrated set harder to use
⇒ Pictures
⇒ Screenshots
⇒ The first pattern form (illustrated) was based on one screen information; whereas the Second one has more information which can cause confusion.
⇒ The use of the illustrations
⇒ The visual aspects
⇒ Visual (pictures)

Narrative Patterns Preference

⇒ Complexity of the information displayed had not much to do with the patterns (Breaking into parts)
⇒ Too much reliance on the Initial example – picture

A18.2 UI-pattern modelling

UI-pattern Model

⇒ Because I could use the structure
⇒ Because it has clear indication of patterns that are needed to recreate the interface
⇒ Figure out quickly what went wrong / needs fixing
⇒ Having used computers enough I have already seen all the patterns in use and the patterns seem to make the job more complicated
⇒ Help me understand UIs better
⇒ I would know how to do it properly
⇒ It showed what order we would need to form all the aspects of the interface
⇒ State information, fields and how things are generally broken up into modular parts that fit the interface makeup you wonder if there were more effective ways to group things
⇒ They would've helped visualize the UI in terms of features and functionality
⇒ We could find which patterns weren't used and debate whether this was correct of not
⇒ Would design it better second time around; provided more time is given with details given beforehand (i.e. read and understand)
⇒ Would help provide ideas for how the UI should be presented

Using the UI patterns when creating the UI-pattern model

⇒ Because it clearly showed how the patterns are applicable and how they relate
⇒ Because it increased my knowledge of patterns from none to a lot in a crash course
⇒ Because then I could see what use the patterns had
⇒ Being able to class UI by parts with names and given specific objectivity makes me wonder if I actually have been doing something similar already
⇒ Didn't attend pattern lecture

- ⇒ Enable me to understand pattern in more details via the two models we have to create
- ⇒ Had not encountered UI patterns before
- ⇒ I didn’t understand the concept at first but then I made sense of it
- ⇒ It gave me an understanding of the fundamentals of patterns by actively participating in the design (construction) of a pattern hierarchy which forced me to understand the basics of each individual pattern that was present
- ⇒ It’s always helpful to get some practical experience
- ⇒ Made you think more about how they inter-related.
- ⇒ My non-existent knowledge of HCI now exists
- ⇒ Put theory into practice

The UI-pattern modelling method

- ⇒ Was not initially very clear what we were expected to produce
- ⇒ Not clear enough that you are only selecting related patterns, and why
- ⇒ Not clear enough what the objective of the exercise is
- ⇒ Helped avoid looking at the same ones too many times
- ⇒ It wasn’t always obvious which patterns belonged together
- ⇒ None, as supervisor directed me to right path/direction.

Steps in Method

Remove

- ⇒ STEP B – Created a bias towards that one example
- ⇒ STEP E – Similar to step D; it is just repetition

Modify

- ⇒ STEP A – Perhaps have a hierarchical map of how the patterns relate [so as] To avoid looking at the same ones many times
- ⇒ STEP A – Similar to step B. could do away with it and start with step B
- ⇒ STEP C – Give all the patterns related and possible hierarchical patterns [because] It wasn’t always obvious which patterns belonged together
- ⇒ STEP E – Continue selecting patterns from those referred to in the previously-selected pattern to construct a hierarchy of related patterns [because] Not clear enough that you are only selecting related patterns, and why – what the objective of the exercise is

A18.3 Communication with Partner

Patterns

- ⇒ Able to decide what was relevant
- ⇒ Allowed a set of standard ways of talking about how the pattern should be defined
- ⇒ Because we could both get the same mind set
- ⇒ Better able to explain point of view
- ⇒ Both self and partner were confused; then we asked for supervisors assistance.
- ⇒ Constructive banter
- ⇒ Gave a place to start discussion, I guess referring the (reverse) modelling process.

- ⇒ In the illustrated patterns we tended to identify them with things we had seen, examples
- ⇒ Lead to discussion on the design patterns – enhancement of knowledge
- ⇒ We could compare patterns and argue for or against each one
- ⇒ We discussed some of the problems and worked through the exercise
- ⇒ We just discussed whether the pattern suited the interface
- ⇒ Whereas we had to add the non-illustrated ones and got a more general sense of their use
- ⇒ Without them we would have lacked terminology to discuss the UI features we were looking at

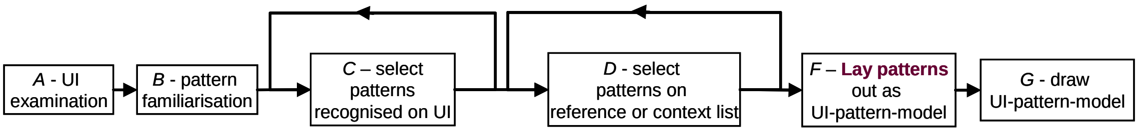
UI-Pattern Modelling Help Design a New UI

- ⇒ These examples were analysis, not design. Would prefer to find the elements needed to design the best UI, while designing it; rather than starting with the elements and trying to assemble them
- ⇒ Can be restricting at times with what pattern can line to what other pattern
- ⇒ Get too caught up in trying to combine several patterns
- ⇒ At first there was a lot of content to cover which seemed daunting at first but once the content was skim read it wasn't so bad
- ⇒ No problems
- ⇒ Perhaps limitation to standard UIs, less room for innovation
- ⇒ Incomplete set of patterns could hinder design
- ⇒ Some functionality or features encountered may not have patterns associated
- ⇒ Too much reading and having to look back and re-read
- ⇒ Very useful but required more hands on to familiarise self with this pattern concept
- ⇒ Thank you for the task and learning curve!

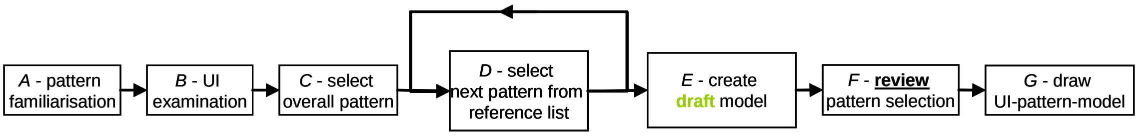
Appendix A19: Study One – Student Derived Methods

The methods derived from the analysis of the image sequences.

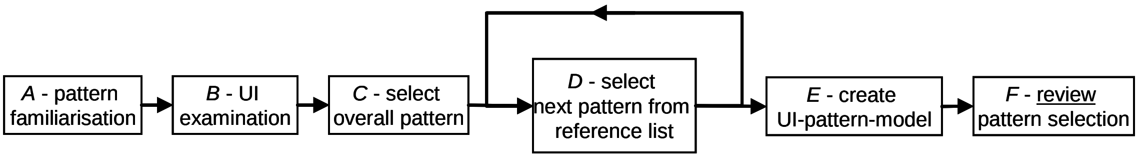
Case 1



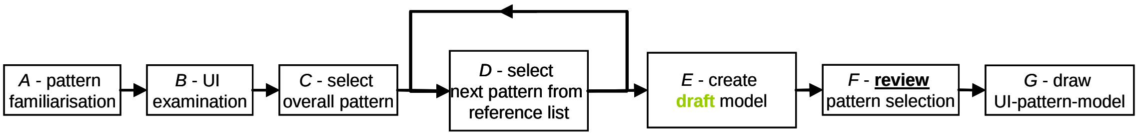
Case 2



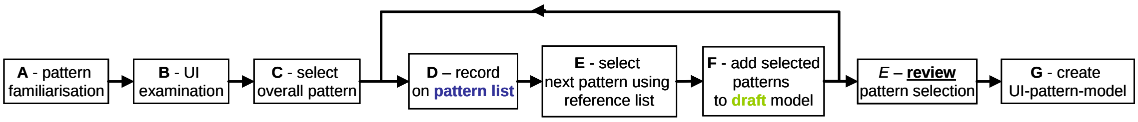
Case 3



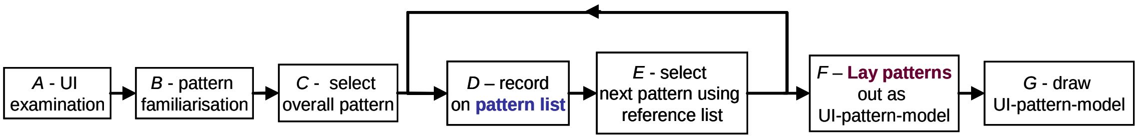
Case 4



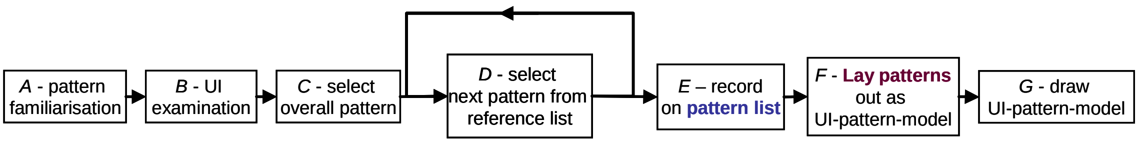
Case 5



Case 6



Case 7



Appendix A20: Study Two – Numeric Data

A20.1 UI-pattern Model Marks

	Appropriate patterns		Potential patterns		Appropriate links		Potential links		Overall	
	Exp1	Exp2	Exp1	Exp2	Exp1	Exp2	Exp1	Exp2	Exp1	Exp2
	1.00	0.89	0.56	0.80	0.50	0.88	0.29	0.78	**0.59**	**0.84**
	1.00	1.00	0.56	0.90	0.50	0.88	0.29	0.78	**0.59**	**0.89**
	0.71	0.83	0.56	0.50	0.50	0.33	0.43	0.22	**0.55**	**0.47**
	0.71	0.83	0.56	0.50	0.50	0.60	0.43	0.33	**0.55**	**0.57**
	0.80	0.64	0.44	0.70	0.75	1.00	0.43	0.33	**0.61**	**0.67**
	0.80	0.75	0.44	0.30	0.75	1.00	0.43	0.33	**0.61**	**0.60**
	0.43	1.00	0.33	0.80	0.83	1.00	0.71	0.89	**0.58**	**0.92**
	0.43	1.00	0.33	0.80	0.83	1.00	0.71	0.89	**0.58**	**0.92**
	0.64	0.80	0.78	0.40	0.42	0.50	0.71	0.44	**0.64**	**0.54**
	0.64	0.67	0.78	0.40	0.42	0.50	0.71	0.56	**0.64**	**0.53**
	0.86	1.00	0.67	0.70	0.50	1.00	0.43	0.67	**0.61**	**0.84**
	0.86	1.00	0.67	0.60	0.50	1.00	0.43	0.56	**0.61**	**0.79**
	0.67	0.82	0.44	0.90	0.60	0.80	0.43	0.89	**0.53**	**0.85**
	0.83	0.82	0.56	0.90	0.80	0.80	0.57	0.89	**0.69**	**0.85**
		0.86		0.60		1.00		0.78		**0.81**
		0.88		0.70		0.86		0.67		**0.77**
		1.00		0.50		0.75		0.33		**0.65**
		1.00		0.50		0.75		0.33		**0.65**
		0.86		0.60		0.67		0.44		**0.64**
		0.86		0.60		0.67		0.44		**0.64**
		1.00		0.40		1.00		0.33		**0.68**
		1.00		0.40		1.00		0.33		**0.68**
		1.00		0.40		1.00		0.33		**0.68**
		1.00		0.40		1.00		0.33		**0.68**
min	**0.43**	**0.64**	**0.33**	**0.30**	**0.42**	**0.33**	**0.29**	**0.22**	**0.53**	**0.47**
max	**1.00**	**1.00**	**0.78**	**0.90**	**0.83**	**1.00**	**0.71**	**0.89**	**0.69**	**0.92**
average	**0.74**	**0.90**	**0.55**	**0.60**	**0.60**	**0.83**	**0.50**	**0.54**	**0.60**	**0.72**
stdev	**0.18**	**0.11**	**0.14**	**0.19**	**0.16**	**0.20**	**0.16**	**0.23**	**0.04**	**0.13**
p T<t	**0.002**		**0.406**		**0.001**		**0.595**		**0.002**	

A20.2 TUIC Modelling Marks

	Appropriate TUIC components	Potential TUIC components	Potential nesting	Appropriate labelling	Overall
	0.56	0.42	0.33	0.45	0.44
	0.60	0.42	0.50	0.62	0.53
	0.60	0.42	0.67	0.62	0.57
	0.60	0.50	0.67	0.64	0.60
	0.67	0.50	0.67	0.67	0.63
	0.67	0.50	0.67	0.67	0.63
	0.67	0.50	0.67	0.78	0.65
	0.70	0.50	0.83	0.78	0.70
	0.73	0.50	0.83	0.82	0.72
	0.80	0.50	0.83	0.83	0.74
	0.80	0.58	0.83	0.87	0.77
	0.80	0.58	0.83	0.88	0.77
	0.82	0.58	0.83	0.88	0.78
	0.83	0.67	0.83	0.88	0.80
	0.83	0.67	0.83	0.89	0.81
	0.86	0.67	0.83	0.93	0.82
	0.88	0.67	0.83	0.94	0.83
	0.88	0.67	1.00	0.94	0.87
	0.89	0.75	1.00	0.94	0.90
	0.90	0.75	1.00	0.94	0.90
	0.90	0.75	1.00	0.94	0.90
	1.00	0.75	1.00	0.94	0.92
	1.00	0.83	1.00	1.00	0.96
	1.00	0.83	1.00	1.00	0.96
min	0.56	0.42	0.33	0.45	0.44
max	1.00	0.83	1.00	1.00	0.96
average	0.79	0.60	0.81	0.83	0.76
stdev	0.13	0.13	0.17	0.14	0.14

A20.3 Patterns Questionnaire Raw Data

Num	Topic	01A	01B	02A	02B	03A	03B	04A	04B	06A	06B	07A	07B	08A	08B	09A	09B	10A	10B	11A	11B	12A	12B
S-01	Clear information	1	1	-1	-1	1	1	1	1	1	0	1	1	0	1	1	1	0	0	0	-1	0	0
S-02	Patterns helpful	1	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	0	-1	1	-1
S-03	Straight forward to select pattern	0	0	0	-1	0	-1	1	1	0	1	1	1	0	0	1	0	-1	1	0	-1	0	-1
S-04	Enjoyed building UI pattern model	-1	1	-1	-1	-1	1	1	0	1	1	1	1	1	-1	1	1	-1	1	-1	-1	1	0
S-05	Finding suitable patterns OK	1	1	-1	-1	0	0	1	1	0	0	1	1	0	1	0	0	1	0	0	-1	-1	0
S-06	Find required knowledge	1	1	0	-1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0	-1	0	1
S-07	Patterns informative	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	0	1
S-08	Diagram helpful	1	0	0	-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1
S-09	Sufficient information	1	1	1	1	1	-1	-1	-1	0	-1	1	1	1	0	-1	0	1	1	0	-1	-1	0
S-10	Easy to find next pattern	1	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	-1	0	-1
S-11	Command of patterns	0	1	0	-1	0	-1	1	1	1	0	0	1	1	1	1	0	0	1	0	-1	0	0
S-13	Liked to use patterns designing UI	0	1	1	-1	1	-1	0	1	1	0	1	0	1	-1	-1	-1	1	0	-1	-1	0	-1
S-14	Straight forward method	-1	1	1	0	1	1	0	1	1	1	-1	1	1	0	0	0	0	1	0	-1	0	0
S-15	Aid UI-concepts discussion	1	1	1	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	0
S-17	Referring back to patterns	-1	-1	1	1	0	1	1	1	0	-1	0	-1	1	-1	0	1	1	1	1	0	1	1
S-18	Patterns easy to understand	1	1	0	0	-1	1	1	1	0	0	1	1	1	1	0	-1	1	1	0	-1	-1	1
S-19	Pattern names clear	1	1	1	1	0	-1	1	1	1	-1	1	1	1	0	0	-1	1	1	-1	-1	-1	1
S-20	Patterns described UI well	1	1	0	1	1	-1	1	1	1	-1	1	0	0	1	1	0	1	1	0	-1	0	1
S-21	Logical organisation	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	-1	1	1	0	1	1	0
S-22	Patterns easy to use	1	1	1	0	-1	-1	1	1	0	1	1	1	1	0	0	-1	1	0	-1	-1	-1	-1
S-23	Number of steps OK	0	0	0	0	-1	-1	1	1	1	-1	0	1	0	-1	-1	0	-1	0	-1	0	-1	-1
S-24	Enough patterns to model UI	1	1	1	0	0	0	0	1	1	1	0	1	1	1	0	0	1	1	0	0	1	1
S-25	Content easy to remember	-1	-1	0	0	1	0	0	0	1	0	0	1	-1	0	1	0	1	0	-1	-1	0	-1
S-26	Pattern names indicate intent	1	1	1	1	0	0	0	0	1	0	1	1	1	1	0	0	0	1	0	-1	-1	1
S-27	Aid modelling discussion	1	1	1	0	0	1	1	1	1	1	1	1	1	-1	-1	-1	1	1	0	-1	1	0
S-28	Method easy to follow	-1	1	-1	-1	-1	1	1	1	1	-1	1	1	0	0	-1	0	1	1	-1	-1	-1	-1
S-29	Examples helpful	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	-1	1	1
S-30	Patterns aid focus	-1	1	1	-1	0	0	0	1	1	0	1	1	0	-1	0	1	1	1	0	-1	0	1
S-31	Understand and act on information	1	0	1	1	1	1	1	1	1	1	1	1	1	-1	1	0	1	1	-1	-1	0	1
S-32	Diagram understandable	1	1	0	1	0	1	1	1	1	1	1	0	-1	0	-1	0	1	1	-1	0	1	-1

A20.4 Patterns Questionnaire Summary Data

Num	Topic	Total	agree	undecided	disagree
S-01	Clear information	9	55%	32%	14%
S-02	Patterns helpful	10	55%	36%	9%
S-03	Straight forward to select pattern	2	32%	45%	23%
S-04	Enjoyed building UI pattern model	4	55%	9%	36%
S-05	Finding patterns OK	4	36%	45%	18%
S-06	Find required knowledge	13	68%	23%	9%
S-07	Patterns informative	16	73%	27%	0%
S-08	TUIC diagram helpful	16	77%	18%	5%
S-09	Sufficient information	3	45%	23%	32%
S-10	Easy to find next pattern	13	68%	23%	9%
S-11	Command of patterns	6	41%	45%	14%
S-13	Liked to use patterns designing UI	0	36%	27%	36%
S-14	Straight forward method	7	45%	41%	14%
S-15	Aid UI-concepts discussion	14	64%	36%	0%
S-17	Referring back to patterns	7	55%	23%	23%
S-18	Patterns easy to understand	8	55%	27%	18%
S-19	Pattern names clear	7	59%	14%	27%
S-20	Patterns described UI well	10	59%	27%	14%
S-21	Logical organisation	14	68%	27%	5%
S-22	Patterns easy to use	3	45%	23%	32%
S-23	Number of steps OK	-5	18%	41%	41%
S-24	Enough patterns to model UI	13	59%	41%	0%
S-25	Content easy to remember	-1	23%	50%	27%
S-26	Pattern names indicate intent	9	50%	41%	9%
S-27	Aid modelling discussion	10	64%	18%	18%
S-28	Method easy to follow	-1	41%	14%	45%
S-29	Examples helpful	17	82%	14%	5%
S-30	Patterns aid focus	6	45%	36%	18%
S-31	Understand and act on information	13	73%	14%	14%
S-32	TUIC diagram understandable	8	55%	27%	18%

A20.5 TUIC Test Results

Question	q1	q2	q3.1	q3.2	q3.3	q3.4	q3.5	q3.6	q4.1	q4.2	q4.3	q5
Answer	d	c	e	f	c	d	a	b	a	b	c	b

Question	q1	q2	q3.1	q3.2	q3.3	q3.4	q3.5	q3.6	q4.1	q4.2	q4.3	q5
	c	a	c	d	c	d	a	b	a	b	b	a
	c	a	c	d	c	d	a	b	a	b	b	a
	c	a	e	f	c	d	a	b	a	b	b	a
	c	b	e	f	c	d	a	b	a	b	b	a
	d	b	e	f	c	d	a	b	a	b	b	b
	d	b	e	f	c	d	a	b	a	b	b	b
	d	b	e	f	c	d	a	b	a	b	b	b
	d	c	e	f	c	d	a	b	a	b	b	b
	d	c	e	f	c	d	a	b	a	b	b	b
	d	c	e	f	c	d	a	b	a	b	c	b
	d	c	e	f	c	d	a	b	a	b	c	b
	d	c	e	f	c	d	a	b	a	b	c	b
	d	c	e	f	c	d	b	b	a	b	c	b
	d	c	e	f	c	d	b	b	a	b	c	b
	d	c	f	f	c	d	e	b	a	b	c	b
	d	c	f	f	c	d	e	b	a	c	c	b
	d	c	f	f	c	f	e	e	a	c	c	c
	d	c	f	f	c	f	e	e	a	c	c	c
	d	c		f	c				a	c	c	c
	d	c		f	c				a	c	c	c
	d	c		f	f				a	c	c	c
	d	c		f	f				a	c	c	c
	d	c										c
	d											c
Num right	20	16	12	20	20	16	12	16	22	15	13	12
Num answered	24	24	24	24	24	24	24	24	24	24	24	24
Percentage correct	83%	67%	50%	83%	83%	67%	50%	67%	92%	63%	54%	50%

A20.6 Exit Questionnaire – Closed Questions

	Proficiency		Communication Aid	UI-Pattern modelling	TUIC modelling
Experienced	2	Very helpful	2	4	3
Reasonable	3	Reasonably helpful	9	5	6
Average or OK	9	Helpful	9	9	6
Not very	5	Not very helpful	3	4	4
Low	3	Unhelpful	0	0	0
		min	0	0	0
		max	9	9	6
		median	3	4	4
		mean	4.60	4.40	3.80
		Std	4.16	3.21	2.49
		skew	0.30	0.15	-0.92
		kurtosis	-2.78	1.69	0.32

Appendix A21: Study Two – TUIC Modelling Case Studies

A21.1 Case 1

	3 mins	**5 mins**	**8 mins**	**11 mins**	**15 mins**	**19 mins**	**23 mins**	**26m**
Artefacts	Pattern list UI UI-pattern model Pattern #1 & #3 TUIC symbols TUIC model	Pattern list UI UI-pattern model Pattern #1 & #3 TUIC model	Pattern list TUIC symbols UI UI-pattern model TUIC model Pattern #3	Pattern list TUIC symbols UI TUIC model	Pattern list TUIC symbols UI TUIC model Pattern #11 & #16	Pattern list TUIC symbols UI TUIC model Pattern #11 & #16	Pattern list TUIC symbols UI TUIC model	Finished
Actions	*Pointing to instruction* Focus on pattern # 3	*Drawing TUIC component*	*Pointing to TUIC component on model* Focus on UI	*Drawing TUIC component* Focus on UI	*Drawing TUIC component* Focus on UI	*Drawing TUIC component* Focus on UI	*Focus on UI*	

Table A21.1 - Observation of case one completing TUIC modelling exercise

After working on this exercise for three minutes these students appear to have completed examining the UI-pattern model and TUIC symbols both of which can be seen on the work space. One student is pointing to the instruction at the top of the worksheet and the other appears to be searching through the pile of patterns probably to find the root node, Pattern 03. These students keep the patterns in ID order for most of the exercise and appear to flick through the pattern language to locate the required pattern with minimal reshuffling or reorganisation. After five minutes they have outlined the overall interaction space for their TUIC models. By fifteen minutes they have outlined the next level of interaction spaces and proceed to add details after requesting an explanation on how to do this. At this point they began to spread some patterns over the workspace.

It is clear that after the initial five minutes these students concentrated on using the ID numbers on the annotated UI to locate the next pattern they would examine. They have added most CAP or navigation symbols to their diagram after nineteen minutes and start adding domain-specific labels.

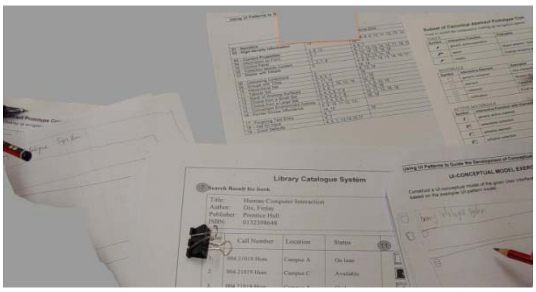


Figure A21.1 - Working from the anotated UI

These students do not sort out the subset of UI patterns making up the UI-pattern model as instructed in step A but followed steps B through to F of the proposed UI conceptual modelling method. At the end they appear to check their model by matching TUIC components with corresponding UI elements.

A21.2 Case 2

	3 mins	5 mins	9 mins	12 mins	15 mins	20 mins	23 mins	27m
Artefacts	UI-pattern model TUIC symbols TUIC model Instructions Pattern collection in clip	UI-pattern model TUIC symbols TUIC model Instructions Pattern #3	UI-pattern model TUIC symbols UI TUIC model Instructions Pattern #1	UI-pattern model UI TUIC model t Pattern #3	UI-pattern model UI TUIC model TUIC symbols Instructions Pattern #3	UI-pattern model UI TUIC model TUIC symbols Instructions Pattern #3	UI-pattern model UI TUIC model TUIC symbols Instructions Pattern #3	Finished
Actions	*Focus on UI-pattern model* Focus on TUIC symbols	*Focus on UI* Focus on TUIC model	*Focus on UI*	*Drawing TUIC component* Focus on pattern #3	*Focus on TUIC symbols*			

Table A21.2 - Observation of case two completing TUIC modelling exercise

After three minutes these students appear to be studying the TUIC symbols and the UI-pattern model, but have not sorted out the subset of patterns matching the model. After five minutes they have shuffled the patterns so that the pattern #3 representing the main interaction space is on top and have drawn the interaction space on their models. At this point the UI-pattern model and the UI have been laid out at the top of the work surface so that both are clearly visible for the rest of the exercise.

After nine minutes they have added the first interaction space for the second level to their TUIC model and at twelve minutes are naming this space. They appear to have been studying the patterns in the interim because these have been reshuffled. The second of the next level of interaction spaces has been added after fifteen minutes.

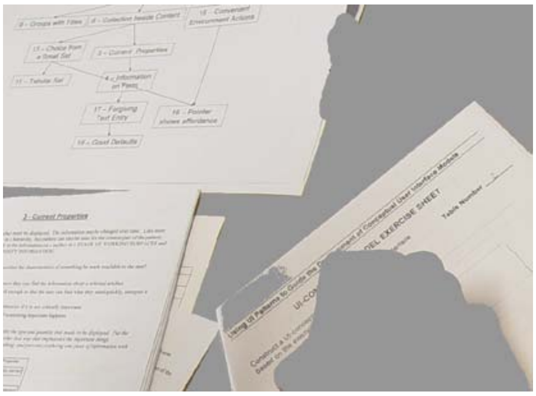


Figure A 21.2 – Adding first interaction space of the second level

By twenty minutes the second level has been completed and some lower-level components have been added. At twenty-three minutes the models are complete and all components have been added and named. The students' may be reviewing their work at this point.

It is unclear what the students have been concentrating on in the last three images because they tidied the work surface when they saw the photographer approaching. They appear to have used both the UI-pattern model and the annotated UI equally when selecting the next pattern to review as both these items were laid out at the top of the work surface for most of the exercise. The TUIC symbols also seemed to be important to these students.

They appear to have followed the steps of the proposed method except for extracting the subset of patterns matching the UI-pattern model as suggested in step A. They do not appear to have updated labels as directed in step F as a number of the names are still generic.

A21.3 Case 3

	3 mins	5 mins	9 mins	12 mins	15 mins	20 mins	24 mins	27m
Artefacts	UI-pattern model Pattern list TUIC model Instructions Pattern collection in clip	UI-pattern model Pattern list UI TUIC model Instructions Pattern #7 **Pattern numbers on UI marked**	UI-pattern model UI TUIC model Instructions Pattern #7	UI-pattern model UI TUIC model Instructions TUIC symbols Pattern #11	UI-pattern model UI TUIC model TUIC symbols Instructions Pattern #16	UI TUIC model Instructions TUIC symbols (under) Pattern #11	UI TUIC model Instructions TUIC symbols UI-pattern model Pattern #11	Finished
Actions	*Focus on UI-pattern model*	*Focus on UI* Pointing to TUIC model diagram on pattern #3	*Pointing to component on UI*	*Focus on TUIC model* Focus on UI-pattern model	*Focus on UI* Pointing to component on UI	*Focus on UI*	*Focus on UI*	

Table A21.3 - Observation of case three completing TUIC modelling exercise

After three minutes these students are studying the UI-pattern model and appear to be comparing it with the UI. Although the TUIC components list is not visible a later image shows that it was on part of the work surface not captured by earlier images. At five minutes one student is pointing to the TUIC model diagram illustrating Pattern 7 and they appear to be comparing this to the UI. The main interaction space has been added to their TUIC model. At nine minutes the UI is central to the workspace and the students have ticked the ids for the patterns they have used. They have completed the interaction space representing Pattern 9 including the details and naming. At twelve minutes they have started trying to model the middle interaction space and asked for help in interpreting the TUIC model diagram for representing the tabular set pattern. At fifteen minutes the UI is again clearly prominent on the work surface and the students are studying pattern sixteen as they are working on the third interaction space. One student is pointing to the UI and possibly marking the id for Pattern 16. At twenty minutes they have returned to the middle interaction space and are back studying Pattern 11 and the UI. At both twenty-four and twenty-seven minutes they are adding the lower level component details that complete their model.

These students predominately relied on the annotated UI to guide their development of their TUIC model, marking the pattern ids off as they completed each part. They added domain-specific names to the components as they developed the model.

The students followed the middle four steps B to E of the proposed method but added domain-specific names as they created the model not at the end, step F. They did not separate out the sub-set of patterns at the beginning, step A but kept the set of patterns in order just turning to each pattern as they required it.

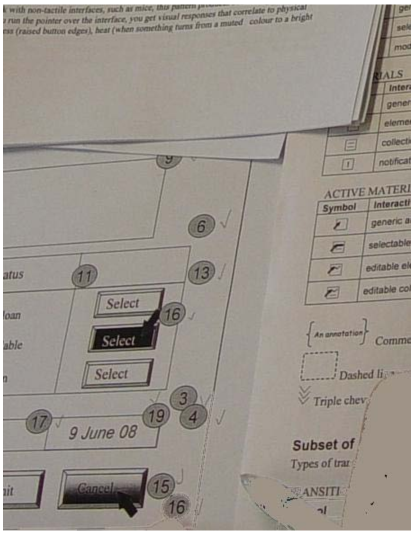


Figure A21.3 - Marking pattern ids off on anotated UI

A21.4 Case 4

	3 mins	6 mins	9 mins	12 mins	16 mins	20 mins	24 mins	27 mins	30m
Artefacts	UI-pattern model UI Pattern list Draft model Instructions TUIC symbols Pattern #1	UI-pattern model UI Draft model Instructions TUIC symbols	UI-pattern model UI Pattern list Draft model Instructions TUIC symbols Pattern #7	UI-pattern model UI Draft model Instructions TUIC symbols Pattern #7	UI-pattern model UI Draft model Instructions TUIC symbols Pattern #13	UI-pattern model UI Draft model Instructions TUIC symbols Pattern #11 & 13	UI-pattern model UI Draft model Instructions TUIC symbols Pattern #13, #11	UI-pattern model UI Draft model Instructions TUIC symbols Pattern #11, #13	Finished
Actions	*Focus on instructions*	*Focus on instructions*	*Focus on pattern list*	*Focus on pattern #7*	*Drawing TUIC component*	*Focus on pattern #11 & #13*	*Holding patterns in hand* Pointing to TUIC component	*Drawing TUIC component*	

Table A21.4- Observation of case four completing TUIC modelling exercise

At three minutes these students have carefully laid out the UI and UI-pattern model at the top of the workspace with the TUIC symbols between them and their pattern sets on either side. They have made a copy of the UI-pattern model as a structured list and now appear to be studying the instructions.

They have erased the copy of the UI-pattern model after six minutes. This image has not captured what they are doing but it is assumed they are studying the UI patterns because they are turned towards where these were placed. At nine minutes they have drawn the main interaction space starting their TUIC model.

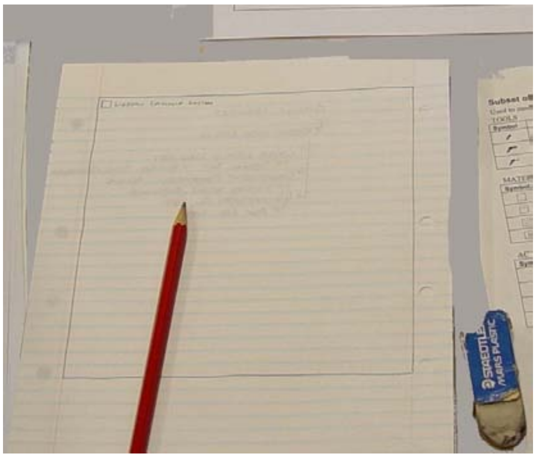


Figure A 21.4 – First attempt rubbed out and restarting with main interaction space

They have added the first of the three interaction spaces on the next level to their model by twelve minutes and have clearly been examining the patterns as a shuffled pile can be seen at the edge of this image. At twelve minutes they appear to be studying pattern number seven and by sixteen minutes they are completing the first of the three second level interaction spaces including adding domain-specific labels to the components. At twenty minutes the students are studying the patterns related to the second interaction space at the next level. At twenty-four minutes they have added the frame for this interaction space and one student is studying the patterns while the other is reviewing the steps in the proposed method. At twenty seven minutes they are adding the last of the three second-level interaction spaces. They are clearly consulting the TUIC symbols as they create their model. By thirty minutes the draft model is complete and they are transferring the model to the worksheets.

The UI and UI-pattern model were clearly visible to both students throughout the modelling exercise but the images capture the student mostly studying the TUIC model diagrams illustrating the patterns rather than concentrating on any other artefact. They kept the patterns in order throughout the exercise, just flipping through utill the patterns required were on top.

These students leafed through the set of patterns to the one required keeping the rest in order. After the false start of just copying the UI-pattern model as a structured list, steps B through E were followed with one variation. In that they choose to create a draft copy first. They also choose to use domain-specific labelling as they added each component rather than at the end. They obviously had to add an additional step where they made final copies of their TUIC model and reviewing their work as they did so.

A21.5 Case 5

	3 mins	6 mins	9 mins	12 mins	17 mins	20 mins	24 mins	28 mins	30 mins
Artefacts	UI Pattern list Instructions	UI UI-pattern model Pattern list TUIC model Pattern #3	UI Pattern list TUIC model Pattern #3	UI UI-pattern model Pattern list Instructions TUIC symbols TUIC model Pattern #3	UI Pattern list Instructions TUIC symbols TUIC model Pattern #3	UI Pattern list UI-pattern model Instructions TUIC model Pattern #3 **Patterns on UI-pattern model marked**	UI Pattern list UI-pattern model Instructions TUIC model Pattern #3, #15 & #6 TUIC symbols	UI Pattern list UI-pattern model TUIC model Pattern #19	UI Pattern list TUIC model Pattern #15, #6 & #16
Actions	*Focus on UI* *Holding patterns in hand*	*Focus on UI-pattern model*	*Drawing TUIC component* Focus on UI	*Pointing to TUIC component on TUIC model* Focus on UI	*Drawing TUIC component*	*Focus on the UI-pattern model* Drawing TUIC component	*Pointing to component on UI*	*Holding TUIC model* Focus on UI	*Drawing TUIC component* **Finished at 32 mins**

Table A21.5- Observation of case five completing TUIC modelling exercise

The UI appears to be the first artefact these students focussed on, but by three minutes one student is beginning to look at the associated patterns. At five minutes focus has moved to the UI-pattern model and the root pattern has been located. At nine minutes the students are drawing the containing interaction space and the UI is clearly the artefact they are using as a focus. By twelve minutes they are working on the first of the three interaction spaces at the next level but still using the UI as they main focus. It is also clear that they have been shuffling through the pile of patterns but are appear to be placing the pile down again in order after locating any patterns of interest.

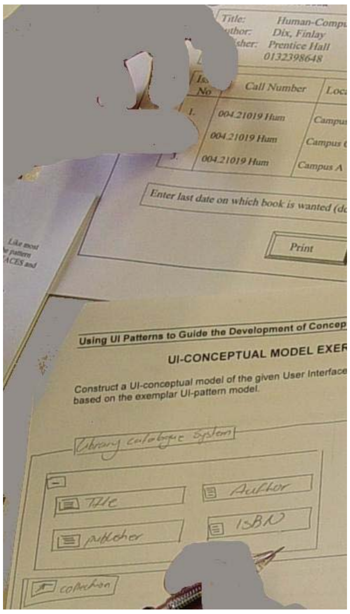


Figure A 21.5 – Apparently discussing how to represent the table

At seventeen minutes these students ask for help and are directed to the UI-pattern model to order their exploration of the patterns. At twenty minutes the students have been focussing on the UI-pattern model and have marked off the first two patterns they have used. They are working on completing the first of the three interaction spaces. By twenty-four minutes the students are again focusing on the UI. They appear to be discussing how to represent the table on the UI as they are pointing to it and are clearly working on the details of the second interaction space at level

two. They also appear to have been viewing
some of the relevant patterns.

The second interaction space has been completed by twenty-eight minutes and it appears that they are reviewing what they have done so far as patterns are fanned across the workspace. At thirty-one minutes these students are still working on their TUIC model completing the final interaction space and take until thirty-two minutes to finish.

These students regularly move their focus to the UI-pattern model and marked patterns on it as they are used them. They also appear to check with the annotated UI at the same time. To begin with they shuffled the patterns and kept them in ID order but later began to spread the patterns on the desktop so that they could see all relevant diagrams.

These students took the longest time to complete the exercise. They effectively swapped steps A and B. Unfortunately the first image is blurred and only partly shows the work surface but it appears the students are studying the UI and TUIC symbols. In the second image they have moved the UI-pattern models into the centre of the workspace. They then appear to follow steps C through E and have used domain-specific labels for some components, but not all because they did not have time to complete step F.

A21.6 Case 6

	4 mins	6 mins	9 mins	13 mins	17 mins	21 mins	24 mins	28m
Artefacts	UI UI-pattern model Instructions TUIC model TUIC symbols Pattern #3 & #4	UI UI-pattern model Instructions TUIC model Pattern #3 & #4	UI UI-pattern model Instructions TUIC model Pattern #3, #15, #11, #9 & #7	UI UI-pattern model Instructions TUIC model Pattern list TUIC symbols Pattern #3 & #8	UI UI-pattern model Pattern list TUIC symbols TUIC model Pattern #11 & #15 **Pattern numbers on UI marked**	UI UI-pattern model Pattern list TUIC symbols TUIC model Instructions Pattern #11 & #17	UI UI-pattern model Pattern list TUIC symbols TUIC model Pattern #11 & #16	Finished
Actions	*Focus on UI* Focus on UI-pattern model Drawing TUIC component	*Pointing to node on UI-pattern model*	*Drawing TUIC component* Focus on UI Holding patterns in hand	*Focus on UI* Pointing to pattern	*Focus on UI* Drawing TUIC component	*Focus on UI* Focus on TUIC model	*Focus on pattern #15 (held in hand)* Focus on TUIC symbols	

Table A21.6- Observation of case six completing TUIC modelling exercise

These students had already added the main interaction space to their diagrams by the time the first image was taken at four minutes. The UI is clearly visible at the top of the work surface and the UI-pattern model is lying between the students. Relevant patterns are also visible. By six minutes they have added the outlines for the first two of the next level of interaction space and one student is pointing to the sub-tree on the UI-pattern model that represents the third interaction space. At nine minutes the UI is clearly in focus and the students are examining patterns while adding the outline of the third interaction space to their models. The UI-pattern model is visible between the students at thirteen minutes and students appear to be studying pattern content. At seventeen minutes the UI is again prominent on the work surface and the students have marked on it the patterns they have used. They are adding details to the middle interaction space. At twenty-one minutes they are still working on the middle interaction space and have added a relevant comment. The third interaction area has been added by twenty-four minutes and the students are focussing on the TUIC symbols, apparently reviewing their models. At this point the first interaction area is incomplete and some components have generalised labels. They consider that they had finished at twenty-eight minutes.

In the beginning stages these students used both the UI and the UI-pattern model to help guide their modelling. Later on, they appear to be more reliant on the UI and began marking on it the patterns they had used. Although they didn't sort out the subset of patterns in the early stages of the exercise they did fan the patterns so a number of relevant patterns could be seen simultaneously.

(A) (B)

Figure A21.6 – Early version using generic names (A) and updated to use domain specific names (B)

These students probably followed the steps of the proposed method reasonably closely. In the first image the TUIC symbols sheet was lying under the UI-pattern model between the students with the UI above them on the work surface indicating the students had probably carried out parts of steps A and B. They then worked through steps C to E labelling some components with generalised labels and some with domain-specific labels. Erasures indicate that step F was applied though not all components were labelled correctly in the submitted models.

A21.7 Case 7

	4 mins	6 mins	9 mins	13 mins	17 mins	21 mins	28m
Artefacts	UI UI-pattern model TUIC symbols TUIC model	UI UI-pattern model TUIC symbols TUIC model	UI UI-pattern model TUIC symbols TUIC model **Patterns on UI-pattern model marked**	UI UI-pattern model TUIC symbols TUIC model	UI UI-pattern model TUIC symbols TUIC model	UI UI-pattern model TUIC symbols TUIC model	Finished
Actions	*Focus on UI* Focus on UI-pattern model	*Focus on TUIC symbols* Pointing to component in list	*Drawing TUIC component* Focus on UI Focus on UI-pattern model	*Drawing TUIC component* Focus on UI-pattern model	*Pointing to TUIC component on TUIC model*	*Drawing TUIC component*	

Table A21.7- Observation of case seven completing TUIC modelling exercise

These students placed the patterns they were using in a pile beside them rather than on the work surface, therefore the patterns they used were not captured in any of the images. At the first image they had laid the UI and UI-pattern model out along the top of the work surface with the TUIC symbols between them and had already competed drawing the enclosing interaction space. By six minutes they had added the first of the three main interaction spaces, and were studying the TUIC symbols. At nine minutes they have marked the patterns they have modelled on the UI-pattern model and are adding the second of the interaction spaces, having already added domain-specific names to the first interaction space but not the detailed contents. At thirteen minutes more pattern nodes on the UI-pattern model have been marked and the students are working on details making up the second interaction space.

By seventeen minutes all pattern nodes on the UI-pattern model have been marked and the students' models now have all three interaction spaces and they have added comments. One student is pointing to a component on their model and they appear to be reviewing their model. At twenty-one minutes one student is capture correcting their model. They have completed their models by twenty-eight minutes.

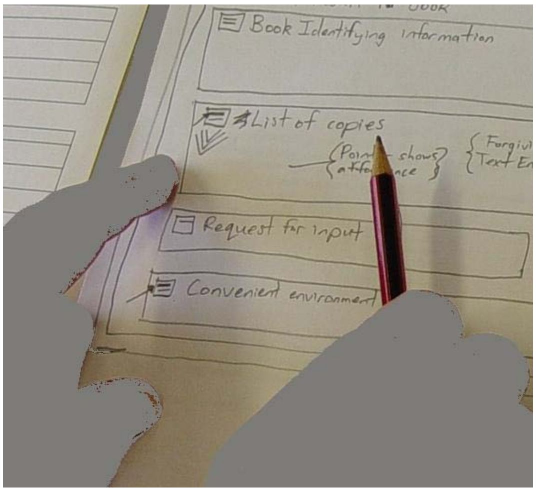


Figure A 21.7 – Apparently reviewing the model

These students clearly focused on the UI-pattern model as they completed their TUIC models. It is assumed that the students just shuffled the patterns to the one they required because they were seen balanced in a pile on a bag beside the students.

It is not clear whether the students followed Step A but they did follow steps B through to E adding domain labels as they created each interaction space. They added a review step at the end of their process.

A21.8 Case 8

	4 mins	6 mins	10 mins	13 mins	17 mins	21 mins	25 mins	28m
Artefacts	UI UI-pattern model TUIC symbols TUIC model	UI UI-pattern model TUIC symbols Pattern list TUIC model Pattern #3 & #7	UI UI-pattern model Pattern list TUIC model Pattern #9	UI UI-pattern model Pattern list TUIC model Pattern #9 & #4 **Patterns on UI-pattern model marked**	UI UI-pattern model TUIC model Pattern #4, #6 & #9	UI UI-pattern model TUIC model Pattern #11, #9 & #4	UI UI-pattern model Pattern list TUIC symbols TUIC model Pattern #17, #6 & #4	Finished
Actions	*Drawing TUIC component* Focus UI-pattern model	*Drawing TUIC component* Focus UI-pattern model	*Pointing to component on UI* Focus UI-pattern model	*Focus UI-pattern model* *Pointing to component on UI*	*Focus on pattern #6* Pointing to TUIC component on TUIC model	*Focus on pattern #11* Drawing TUIC component	*Focus on pattern #17* Drawing TUIC component	

Table A21.8- Observation of case eight completing TUIC modelling exercise

In the first image the students have laid the UI and the TUIC symbols at the top of the work space and have the UI-pattern model between them. They are drawing the enclosing interaction space. At six minutes they have started adding the second interaction space and seem to be focused on the UI-pattern model. At both 10 minutes and 13 minutes one student can be seen pointing to components on the UI and they appear to be discussing how to represent the details of the second interaction space. They have also marked pattern nodes on the UI-pattern model. At seventeen minutes they are now studying Pattern 6 and are pointing to their current representation of the second interaction space, obviously reviewing their model so far. By twenty-one minutes they are adding detail to the second interaction space concentrating on Pattern 11. At twenty-five minutes the students are completing the third interaction space and they have completed their models by twenty-eight minutes.

These students predominately relied on the UI-pattern model to guide the development of their TUIC model, which shows pattern nodes marked. They added domain-specific names to components as they developed the model.

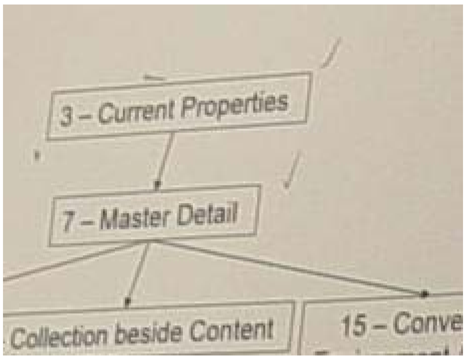


Figure A 21.8 – Marking progress on the UI-pattern model

Patterns were placed across the work space as they were required. These students followed steps B through E. They used domain-specific names to label components as they developed their models rather than at the last step.

A21.9 Case 9

	4 mins	7 mins	10 mins	13 mins	18 mins	22 mins	25m
Artefacts	UI UI-pattern model TUIC symbols TUIC model Instructions Pattern #3 **Matching TUIC components listed on UI-pattern model**	UI UI-pattern model TUIC symbols TUIC model Instructions Pattern #3, #9 & #8	UI TUIC model TUIC symbols Instructions	UI UI-pattern graph TUIC model Instructions TUIC symbols Pattern #3	UI UI-pattern graph TUIC model Instructions TUIC symbols Pattern #3	UI UI-pattern graph TUIC model Instructions Pattern #3 & #9	Finished
Actions	*Drawing TUIC component*	*Focus on TUIC model* Focus on TUIC symbols	*Pointing to component on UI*	*Focus on UI-pattern model*	*Focus on TUIC model*	*Focus on TUIC model*	

Table A21.9 - Observation of case nine completing TUIC modelling exercise

In the first 4 minutes these students have already had time to create a list of CAP and navigation symbols which they consider will be used to describe the UI on the side of the UI-pattern model. They have also labelled the relevant components on the UI with the labels ‘overview’ and ‘details’. One student is adding a component to their model while the other is examining the list of TUIC symbols. Both the UI and UI-pattern model are prominent on the work surface. At ten minutes they appear to be matching a component on their TUIC model with one on the UI. At this point it is clear that these students are creating an ordered list rather than a diagram for their TUIC model. By thirteen minutes they have modelled the first interaction space and are completing the second one. At seventeen minutes they have begun modelling the third interaction space. The UI, UI-pattern model and TUIC symbols have remained in view throughout the exercise. At twenty one minutes they are completing their models and had finished by twenty-two minutes.

These students appeared to have used both the UI and the UI-pattern model to help guide their modelling throughout the exercise. Their use of the patterns is not clear as in most images these are not shown. They added domain-specific names to the components as they developed the model.

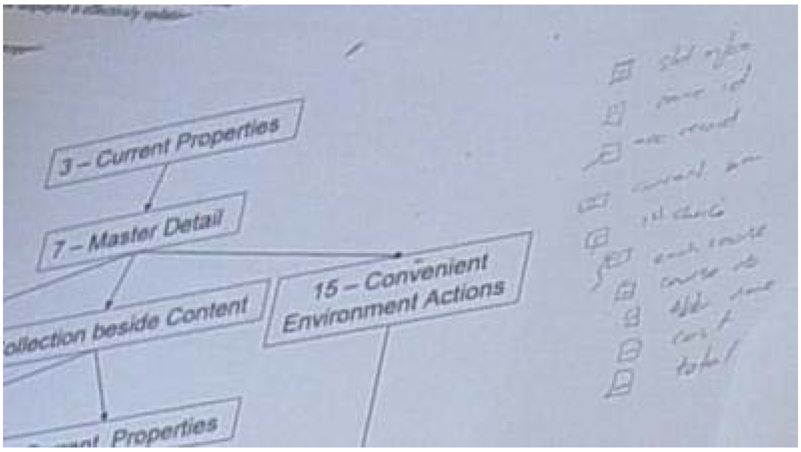


Figure A 21.9 - Listing TUIC components on UI-pattern model

It appears that these students did follow the proposed method for steps A through to E. They recorded the CAP and navigation symbols shown in the diagrams illustrating the patterns on the UI-pattern model as they examined each pattern and. They appear to have identified which parts of the UI these represented. Next they are seen still studying the TUIC symbols as they start their modelling exercise. Rather than add domain-specific labels as step F these students choose to use domain-specific names to label components as they developed their models.

A21.10 Case 10

	4 mins	**7 mins**	**10 mins**	**14 mins**	**18 mins**	**22 mins**	**26m**
Artefacts	UI UI-pattern model Pattern list TUIC model Pattern #3 Pattern #1	UI UI-pattern model Pattern list TUIC model Instructions Pattern #8	UI UI-pattern model TUIC model Drawing TUIC component Pattern #11 Pattern #9	UI UI-pattern model TUIC model Focus on pattern Pattern #9 Pattern #6 Pattern #3	UI UI-pattern model TUIC model Pattern #19 Pattern #11 Pattern #3 Pattern #6	UI UI-pattern model TUIC model Pattern #11 Pattern #3 Pattern #6	Finished
Actions	*Focus on pattern* Focus on instructions Pointing to the reference section pattern #3	*Focus on pattern #8* Focus on UI	*Focus on pattern #11* Focus on TUIC model	*Focus on UI-pattern model* Drawing TUIC model	*Focus on TUIC model* Pointing to TUIC model diagram on pattern 19	*Focus on the UI* Pointing to TUIC component on TUIC model	

Table A21.10 - Observation of case ten completing TUIC modelling exercise

The UI-pattern model is placed at the top of the work surface while one student is studying the reference section of Pattern 3 and the other appears to be checking out the instructions in the first image. One student is studying Pattern 9 while the other student is studying the UI. The UI-pattern model is still clearly visible between the students at the top of the work surface and they have created the containing interaction space and completed the first of the three main spaces making up their TUIC model after six minutes has elapsed. By ten minutes they have started drawing the second interaction space and are studying patterns nine and eleven. The UI is now placed on top of and over lapping the UI-pattern model. They are completing the details of the second interaction space at thirteen minutes and one student appears to be searching for the next pattern to examine. At seventeen minutes the students appear to be considering how to indicate on their TUIC model the good defaults pattern and one student appears to be pointing to its TUIC model diagram. By twenty-two minutes the third interaction space is being added with the UI lying in a prominent position.

Figure A 21.10 - Most images clearly show TUIC model diagram on patterns being consulted

These students appear to have initially concentrated on the UI-pattern model and it was visible throughout the exercise. But in the finishing stages the UI seemed to be more prominent. Their main focus though was on the TUIC model diagrams illustrating each of the patterns. They added domain-specific names to components as they developed the model.

These students followed steps A through E of the proposed method. In the first image the UI-pattern model is prominent and they are studying the TUIC model diagram of Pattern 3. In the second image the TUIC symbols can be seen between the students beneath the TUIC model they are working on. As the students worked on steps C, D and E they labelled components with UI domain-specific names. It is not clear what the students did after completing the third interaction space and moving onto the next exercise at the end.

A21.11 Case 11

	4 mins	7 mins	10 mins	14 mins	18 mins	22 mins	26m
Artefacts	UI UI-pattern model Pattern list TUIC model Pattern #3 Pattern #1	UI UI-pattern model Pattern list TUIC model Instructions Pattern #8	UI UI-pattern model TUIC model Pattern #11 Pattern #9 **Patterns on UI-pattern model marked**	UI UI-pattern model TUIC model Pattern #9 Pattern #6 Pattern #13	UI UI-pattern model TUIC model Pattern #19 Pattern #11 Pattern #13 Pattern #6	UI UI-pattern model TUIC model TUIC symbols Pattern #17 Pattern #19	Finished
Actions	*Focus on pattern* Focus on UI-pattern model Pointing to the table on UI	*Focus on pattern on pattern #8* Focus on UI	*Focus on pattern #11* Focus on TUIC model	*Focus on UI-pattern model* Focus on TUIC model	*Focus on TUIC model* Pointing to pattern node on UI-pattern model	*Focus on TUIC model* Pointing to TUIC component on TUIC model	

Table A21.11 - Observation of case eleven completing TUIC modelling exercise

These students started work very quickly and by the time the first image was taken they had already completed the framing interaction space as well as the first and third interaction spaces on a draft copy of the TUIC model. They have shifted their attention to the second interaction space and at seven minutes are studying Pattern 8. Both the UI and the UI-pattern model are placed at the top of the work space between the students. By ten minutes they had added the second interaction space and started on adding details. They have marked the patterns they had used on the UI-pattern model. At fourteen minutes the students have not added any more details to their TUIC model and they appear to be studying the relevant patterns. By eighteen minutes more details have been added to their model and one student is captured pointing to a node on the UI-pattern model. At twenty-two minutes the draft model is complete and the students appear to be reviewing their work before transferring it to the exercise worksheets.

The UI-pattern model and pattern content appear to have been important to these students although the UI was focussed on in the earlier stages. The patterns required at any one time were scattered across the work surface so that several could be seen at one time. A second pile of patterns, apparently not of current interest, can sometimes be seen off to one side. These students labelled components with domain relevant names as they created their model.

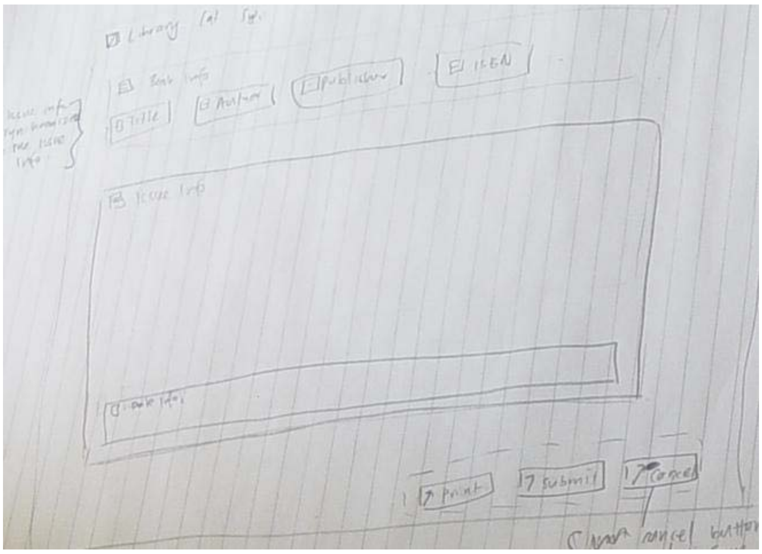


Figure A 21.11 – Partially TUIC model with the easier interaction spaces completed

Because these students started quickly there is no evidence as to whether they followed steps A or B. They have followed steps C, D and E. They choose to do the easier interaction spaces first, leaving the most complex till last. They added a review step at the end.

A21.12 Case 12

	4 mins	8 mins	11 mins	14 mins	19 mins	23 mins	26 mins	29m
Artefacts	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #3	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #7	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #7 & #6	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #13, #18 & #19	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #16 & #4	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #16 & #4	UI UI-pattern model TUIC symbols Pattern list Instructions TUIC model Pattern #7	Finished
Actions	*Focus on pattern #3*	*Focus on UI* Focus on pattern #7 Pointing to component on TUIC symbols list	*Focus on UI* Focus on pattern #6 Drawing TUIC component	*Focus on patterns #13 & #19* Drawing TUIC component	*Focus on UI* Pointing to solution section on pattern #4 Pointing to TUIC component on TUIC model	*Focus on UI* Drawing TUIC component	*Drawing TUIC component*	

Table A21.12 - Observation of case twelve completing TUIC modelling exercise

These students are studying pattern content at four minutes. The UI, UI-pattern model and TUIC symbols are placed in prominent positions on the work surface indicating that they have been examined earlier. At eight minutes the UI is prominently placed between the two students, with one student studying the TUIC model diagram on a pattern and the second pointing to a symbol on the TUIC component list. Their model shows the containing interaction space with the frames for the three interaction spaces. The third space has been completed and they are adding details to the first one. One student is focused on the TUIC model diagram of Pattern 7. By eleven minutes they are adding details to the second interaction space. At fourteen minutes and nineteen minutes the students are adding details to this interaction space and can be seen consulting relevant patterns. The UI is prominent between the students throughout this period. At twenty-three minutes it is clear that the students have erased and changed some parts of their model, apparently after reviewing their work. At twenty-six minutes they are adding comments to their model and they have finished by twenty-nine minutes.

These students used the UI to determine which patterns they would examine next. In the earlier images the students kept the patterns clipped together in order and just turned to the pattern they were studying. Once they began to model the details of the second interaction space they began to sort the patterns and had a number of relevant patterns spread across the work space.

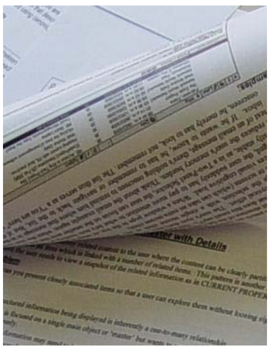


Figure A 21.12 - Flicking through clipped patterns

It is assumed that these students at least partially followed steps A and B before they began creating their model. They followed steps C, D and E. They reviewed some aspects of their model in the later stages of the exercise but apparently not names because they used a mixture of generic and domain-specific names for TIUC component labels.

Appendix A22: Study Two – Exit Questionnaire Responses

To ensure anonymity the comments have been amalgamated and sorted alphabetically.

A22.1 UI-pattern Patterns

Justification for most useful pattern section

⇒ Another perspective (Problem context)
⇒ Basis of further design (Problem)
⇒ Can just follow the diagram it is easy reference can send directly to (next) pattern to look at (Diagram reference)
⇒ Could visually see textually understand what the pattern was and what scope it could be implemented (Examples discussion reference)
⇒ Easier to understand and knowing to do next (Diagram reference)
⇒ Helped link up the different patterns (Context reference examples)
⇒ I could look at and recognise them (examples)
⇒ It gave us a quick reference point to be able to complete the desired tasks in minimal timeframe (diagram)
⇒ It helped me understand the pattern (Discussion diagram)
⇒ It helped relate an unfamiliar concept to things commonly encountered (example)
⇒ It showed what it was about (name context)
⇒ It was good to see the pattern in use (Reference, examples)
⇒ Quick (diagram)
⇒ Quick to understand pattern (Examples)
⇒ So I can look at it and be able to recognise them (examples)
⇒ Very helpful (Diagram)
⇒ Visual aspects are clearer (Context, diagram)
⇒ Visual data is easy to understand (Diagram examples)

Justification for least useful pattern section

⇒ Can continue without discussion (Discussion)
⇒ Did not have time to read it and understand (discussion)
⇒ Didn’t make a lot of sense I couldn’t relate to the pattern (Forces)
⇒ Didn’t need to read it (discussion)
⇒ Just didn’t find it relevant (forces)
⇒ Might be a way of reference (Diagram)
⇒ Not enough time to read it (discussion)
⇒ The names were unclear without reading the description (name)
⇒ Time consuming (discussion)
⇒ Too hard to understand (Forces)
⇒ Too long (discussion)
⇒ Too long to read (discussion)
⇒ Too much information (discussion)

Relevance of Diagram

Yes

- ⇒ Because it gave us a visual aid to what we were supposed to come up with
- ⇒ Gave you a general idea
- ⇒ Helped visually picture how to implement
- ⇒ It gave us a quick reference point to be able to complete the desired tasks in minimal timeframe
- ⇒ It gave you a general idea
- ⇒ It helps to visualise things
- ⇒ Kept on track
- ⇒ Once you understood gave basic overview of what looking for/at
- ⇒ Very limited
- ⇒ Visual clues

No

- ⇒ The diagrams alone doesn't quite help to get the big picture of what its about
- ⇒ I read the context and looked at the examples only most times

General Comment

- ⇒ Funny and interesting
- ⇒ The examples and context and identifiers were the parts I focused on
- ⇒ The list of available patterns would have been easier to refer to if they were grouped
- ⇒ The list with numbered reference was very good
- ⇒ Time was too short
- ⇒ Very interesting – need to learn more

A22.2 UI-pattern modelling

Using the UI patterns when creating the UI-pattern model

- ⇒ Actual activities help me remember
- ⇒ All information provided when knowledge lacking
- ⇒ Because I had no previous knowledge
- ⇒ Found it confusing as not enough time to look at all the info
- ⇒ Gave a standard and background knowledge for non-designers
- ⇒ Gave me a chance to use what knowledge I currently have
- ⇒ Good for remembering what detail needs to be specified
- ⇒ Good to see pattern examples
- ⇒ It allowed me to see a whole new world
- ⇒ Now I can reasonably create a UI pattern model for other user interfaces
- ⇒ Showed examples
- ⇒ Too quickly done

The UI-pattern modelling method

- ⇒ Interesting exercise once I had understood what needed to be done. Good working with a partner.

⇒ Like to do more
⇒ More time would be nice
⇒ Some people don't know foreign language from the pattern can get the what is talking or information
⇒ Thanks for the exercise

Steps in Method

Remove

⇒ More explanation required

Modify

⇒ STEP A – More time
⇒ STEP A – Need more time to become familiar

A22.3 TUIC Modelling

Using the UI patterns when creating the TUIC model

⇒ Good to see pattern examples
⇒ I can now reasonably be able to create a UI conceptual model for other user interfaces.
⇒ I used them
⇒ It clarified what knowledge we had attempted to do in the first exercise
⇒ Just provided a new way of doing UI hence more knowledge on how to design good UI.
⇒ More of an understanding of what we were doing
⇒ Need understanding of patterns to understand how they fit together
⇒ Never done this before
⇒ Step by step referencing

The TUIC model building method

⇒ Very good tut and very useful.
⇒ Enjoyed doing the exercises

Steps in Method

No steps identified just two general comments

⇒ It's clear enough to understand too
⇒ Needed more explanation

A22.4 Communication with Partner

Patterns

⇒ Able to point to examples and compare physical evidence.
⇒ Another point of view
⇒ Because it made it easier to get the exercise done.
⇒ Because then I knew that he would know what I was talking about.

- ⇒ Can discuss issues and get other perspectives on choosing a pattern.
- ⇒ Common points of reference and other views to think about.
- ⇒ Explaining while answering.
- ⇒ Good reference.
- ⇒ Had difficulty understanding and taking in all the information.
- ⇒ It gave us a common point of reference.
- ⇒ It meant we had to discuss more to understand it.
- ⇒ It’s hard to work by myself with a set time.
- ⇒ There were really good examples and related to the context.
- ⇒ They have insight into formal design methods.
- ⇒ They provided an example and related to the context.
- ⇒ You actually have something rely on and get to talk about.

Appendix A23: Study Three – Numeric Data

A23.1 UI-pattern Model Data

	Participant	One	Two	Three
Patterns	valid	22	12	12
	not valid	0	0	0
	missed	2	7	8
	%appropriate patterns	100%	100%	100%
	% potential patterns	92%	63%	60%
Structure - links	appropriate	20	11	10
	inappropriate	1	0	0
	missed	0	0	1
	% appropriate links	95%	100%	100%
	% potential links	100%	100%	91%
	overall score	97%	91%	88%

A23.2 TUIC Model Data

Participant	One	Two	Three
valid	18	19	19
not valid	0	0	1
missed	3	2	4
% appropriate TUIC components	100%	100%	95%
% potential TUIC components	86%	90%	83%
general/unnamed	0	0	0
specific	18	19	19
% appropriate labels	100%	100%	100%
nesting (4)	4	4	4
% potential nesting	100%	100%	100%
overall score	96%	98%	94%

A23.3 Patterns Questionnaire

Participant	One	Two	Three
The information presented in these UI patterns is quite clear.	1	0	0
These UI patterns are helpful when modelling user interfaces.	1	1	1
Usually I found it straight forward to find which pattern to select next.	1	0	1
I enjoyed using these patterns to build a UI pattern model of the user interface.	1	1	1
It was not difficult to find suitable patterns to model the user interface.	1	1	0
Most of the time I could locate the information I needed in these patterns.	1	1	1
The content described in these patterns is very informative.	1	1	1
The diagram helped me identify the essential parts of a pattern's solution.	1	1	0
The patterns helped me focus when building the TUIC model	1	1	1
It is relatively easy to move from one pattern to the next one when selecting patterns to describe a user interface.	1	0	1
I felt in command of these patterns when I used them to create the UI-pattern model.	1	-1	1
Creating the TUIC model using TUIC components was relatively easy.	1	1	1
I liked using the UI-patterns when I design a new user interface.	1	1	1
Creating the UI-pattern model could be performed in a straight forward manner using these UI patterns.	1	1	0
The method for creating a TUIC model does not have too many steps	0	0	1
There is sufficient information in each pattern.	1	1	1
I enjoyed using these patterns when building the TUIC model.	1	1	1
The UI patterns are easy to understand.	1	1	0
I found the pattern names clear	1	0	-1
The patterns selected to model the user interface described it quite well.	1	1	1
The organisation of the information in each of these patterns seems quite logical	1	1	0
Learning how to use these patterns was not difficult.	1	0	1
There are not too many steps required to create the UI pattern model.	1	1	1
The set of patterns provided was adequate for modelling the UI in the exercise.	1	1	1
It is easy to remember the information in these UI patterns.	1	1	-1
The pattern names indicated each pattern's intent.	1	0	1
Building the TUIC model could be performed in a straight forward manner using the UI patterns.	1	1	1
Creating the UI-pattern model was easy	1	-1	0
The examples helped me understand the information in these UI patterns.	1	1	1
The patterns helped me focus my thoughts when building the UI-pattern model.	1	1	1
I can understand and act on the information provided in these patterns.	1	1	1
I could understand the diagram illustrating the solution.	1	1	0

A23.4 Exit Questionnaire

Factor	Participant One	Participant Two	Participant Three
Developing a UI-pattern Model	Extremely helpful	Very helpful	Extremely helpful
Patterns guide TUIC modelling	Extremely helpful	Extremely helpful	Extremely helpful
Suitability of TUIC modelling	Extremely helpful	Very helpful	Very helpful
Patterns impact on communications	Extremely helpful	Extremely helpful	Helpful
UI-pattern modelling on communications	Extremely helpful	Very helpful	Extremely helpful
TUIC modelling on communications	Very helpful	Extremely helpful	Very helpful

Appendix A24: Study Three – Case Study Descriptions

A24.1 Participant One

Participant One was completing a PhD in Engineering. Participant One was introduced to UI design and development in the undergraduate programme six and a half years ago. Since then experience developing UIs has been associated with research projects. UI experience was self-rated by Participant One as being novice to average. The main approach to UI design was through User Centered Design supported by Use Case development and lo-fi prototyping. This participant had been introduced to Canonical Abstract Prototyping as part of their degree programme and had been the subject of a UI-pattern modelling pilot study. They self-rated as not proficient in both these modelling methods.

Participant One reported that the main tools used for UI development were PowerPoint for lo-fi prototyping and the visual toolbox components of the Delphi integrated development environment (IDE).

UI-Pattern Modelling

Participant One took two hours and twenty-six minutes to complete the UI-pattern modelling session. Fifty-nine minutes were taken up by the demonstration and subsequent familiarisation with the exemplar used in the workbook. Discussion revolved around using the context and reference sections for selected patterns to locate further patterns.

After studying the requirements documents for the exercise Participant One asked for confirmation of the importance of the interaction spaces and actions lists. UI-pattern modelling took forty-six minutes. The two main discussion topics were first about how patterns ‘08 Cascading collections’ and ‘07 - Collections beside Content’ represented the interactions space ‘Finding a Course’ and secondly about the types of data that could be represented using the patterns: ‘03 – Current Properties’, ‘04 - Information on Form’, ‘05 - Control Panel’ and ‘06 - Master with Details’.

Participant One was very clear about how patterns were used as part of UI-pattern modelling saying “*The name because it gave us a quick reminder of what the patterns was for (i.e. its purpose). The context and reference sections because we used these for building the tree (upwards and downwards). The diagram and examples because they* *provided a quick way of understanding in what situations the pattern can be applied."* The least useful were seen as "*The Forces and Discussion sections because when you become familiar with the patterns you don't need to look at these anymore*". TUIC model diagrams were rated very positively and were seen to help to "*…more quickly visualise the solution and decide whether the pattern was suitable or not*". They suggested that a way to sort through patterns more easily would be helpful but did not provide any suggestions as to how this might be accomplished.

Participant One considered the UI-pattern modelling process was effective and that it could be very helpful when modelling a complex UI but considered it was time consuming. They liked the way the patterns provided lots of suggestions on how a problem could be solved and thought that creating the UI-pattern model had the potential to reduce later changes. In the final discussion they commented that modelling from requirements was different from their experience in the earlier pilot where the modelling was for an existing UI.

The UI-pattern model Participant One created scored ninety-seven percent, missing just a minor details. They chose to use chains of patterns and did not model interaction spaces for displaying a hotel room's information, so that the room's identity would remain visible and similarly the model for a client's reservation would not show the client's name as the different parts of the transaction were completed. They considered using patterns and developing a UI-pattern model when there are non-professionals in a UI design team "*would speed up the process of designing a good UI*".

TUIC Modelling

It took Participant One an hour and thirty-three minutes to complete the third session. The introduction and demo took twenty-eight minutes. They examined the patterns used to guide the development of the example TUIC model and asked for explanations of how a number of the diagrams illustrating solutions translated to the developing prototype; for example, how the TUIC model diagram for Pattern '13 – Tabular Set' manifested itself in the TUIC model. The discussion covered the alternative diagrams used to illustrate some patterns such as Pattern '12 – Choice from a small set' which lead to an explanation of active materials. Participant One also asked for an explanation of how the representation for the 'Current programme' related to the chain of patterns seen in the UI-pattern model.

Only two minutes were required for revising the requirements for the exercise and the exemplar UI-pattern model. Participant One took only eighteen minutes to complete the TUIC model. During that process the discussion was mostly of the form where the participant asked for confirmation of TUIC model diagram interpretations and their inclusion in the TUIC model. At the end, they reviewed their TUIC model and made the comment that the interaction space for “room selection” could be modelled by including more table details.

Participant One identified the Name, Solution and associated Diagram as the most useful sections of a pattern when completing the TUIC model and commented that the other parts were not relevant to this activity. They considered the diagrams were very helpful and commented “*because the patterns had the CAP symbols on the diagram I didn’t need to try and remember what one I should use*”. They also found the diagrams helped visualise how to model the different parts of the prototype.

Participant One found the steps in the method quite wordy but did not put forward any alternatives. The demonstration example was seen as being very helpful and the suggested providing the example was most helpful. In the follow-up discussion they made the comment that using the UI-pattern model made TUIC Modelling much easier than working directly from use cases. This non-pattern approach had been used as part of a class assignment four years earlier. Reponses to the Patterns Questionnaire were very positive with no disagreements and only one undecided response that queried the number of steps in the TUIC Modelling method.

The TUIC model created from the exemplar UI-pattern model provided scored ninety-six percent (Appendix A29). Participant One did not add the detail for the table representing the room selection list but recognised this had not been completed. TUIC Modelling was rated as extremely helpful commenting it “*allows you to more easily discuss with and explain to others your ideas for how to design the UI*”.

Communication

Participant One thought using patterns with a UI design team that included non-professional members would be extremely helpful commenting “*Team members will have a common terminology and can therefore more easily and quickly understand each what other members mean. Team members can more easily point out why a solution might be suitable or unsuitable, or more effective or not, by referring to a pattern.*”

They also rated UI-pattern modelling as being extremely helpful in aiding communication between all team members commenting that it would allow them to "*quickly see what the proposed interface will offer and can therefore more easily and efficiently identify things that should be modified or things that are missing*" making *"communication easier and faster*".

Participant One rated using patterns and TUIC Modelling with non-professionals as very helpful commenting that as a member of the design team it would be easier to "*Team members can more easily argue their point of view about modelling things in the TUIC model when they have the UI patterns solution (text and diagram) sections to help support their argument*".

A24.2 Participant Two

Participant Two is an information technology professional who is currently working on maintaining and developing a large departmental website. Participant Two has had eighteen years experience with desktop application development. Participant Two self-rating as an expert in UI development reporting they are currently involved in UI development on a daily basis. The main approach is ad-hoc with a strong emphasis on hi-fi prototyping using a variety of desktop based IDEs including: Visual Basic 6, 7 & 2000, FrontPage, DreamWeaver with Photoshop, MS Access, Visual Studio, Visual Interdev and Expression Web. Participant Two had been introduced to UI-patterns in the early stages of this research but had not used them to create a new user interface design. Follow-up discussion also identified familiarity with task modelling using Use Cases.

UI-Pattern Modelling

It took Participant Two only one hour thirty-seven minutes to complete the UI-pattern modelling session. A minimum of time was spent studying both the introduction to the session and the requirements documentation for the exercise. The discussion associated with the demonstration session focussed first on differentiating between patterns ‘08 Cascading collections’ and ‘07 - Collections beside Content’ and secondly on the chain of patterns ‘03 – Current properties’ linked to ‘06 – Master with Details’ linked to ’10 – Stack of Working Surfaces’.

Participant Two examined the exemplar UI-pattern model and the associated UI at the start of the modelling exercise. Then the example illustrations were used to quickly, sort out a set of patterns that related to visual components selected as suitable for meeting the UI specification. At this point they tried to link just these patterns together without consulting either the reference or context sections of the patterns. The researcher intervened and reviewed the example UI-pattern model using a bottom-up approach for linking patterns together. They made the comment that they had difficulty conceptualising UI ideas. A requested was made to help in linking the lower level patterns so as to find a higher level or parent pattern.

Participant Two’s identification of the name, context and examples section as being the most useful was consistent with the bottom-up approach taken to creating the UI-pattern model. The context section was identified as important for determining which patterns to consider next. The bulk of the text was identified as being the least useful, and a verbally comment indicated they preferred using visual information. They did not think the TUIC model diagram helped them identified the essential elements of the solution commenting in relation to the text and diagram that they "*required more figuring out than images [examples] did but provided essentially the same information*". Still Participant Two liked using patterns saying "A g*reat way of piecing together a solution*".

Overall the experience of using patterns was positive, as gauged from responses to the Patterns Questionnaire, with only negative responses to two statements. Participant Two did not feel in command of the patterns when using them and did not find UI-pattern modelling easy. These responses confirmed statements made that conceptualisation and abstract UI modelling was difficult.

Participant Two did not suggest any modifications for the proposed method but did suggest that an explanation on the difference between using it as a top-down approach in comparison to using it as a bottom-up approach would be helpful. They saw the UI-pattern modelling as "*a multi-dimensional map that allows and directs creation of a solution specific to requirements*". They liked the flexibility of the method but reiterated their personal difficulty with abstracting UI ideas.

Participant Two created scored ninety-one percent but was still a minimalist UI-pattern model (Appendix A28). The main omission was not modelling the actions associated with interaction spaces. The model for the transaction interaction space would not enable the client's name to be visible as the different parts of the transaction were completed. They considered using patterns and developing a UI-pattern model would help non-professionals in a UI design team because "*they are building an understanding of the design process and logic which will also help them understand when the final product is different to anticipated solution*".

TUIC Modelling

Participant Two took one hour and thirty-four minutes to complete the TUIC modelling session. At the beginning of the demonstration Participant Two examined the exemplar UI-pattern model carefully then watched and listened without interruption other than to confirm their understanding of a particular point and to indicate that the demonstration should continue.

At the conclusion of the demonstration Participant Two took seven minutes to become re-acquainted with the UI requirements documentation for the exercise. At the beginning of the TUIC modelling exercise the enclosing context space was quickly drawn on the exercise sheet but then they asked for clarification on dealing with the login process. This resulted in a discussion of the appropriate Use Case and its relationship to the exercise. TUIC Modelling resumed after ten minutes, interspersed with a series of question to confirm actions or to clarify interpretation of the diagrams. They initiated marking the UI-pattern model by ticking patterns as they were added to the prototype. Some time was taken up with discussing how to model a ‘selection from a list’ that could be organised as a table of values. The other major discussion covered how to model chains of patterns. The demonstration exemplar was used to illustrate how such chains could be dealt with. The researcher assisted by finding appropriate patterns, when Participant Two tried to remember a diagram rather than re-checking it. In total the TUIC model took Participant Two forty-two minutes to complete.

The Name, Solution and associated Diagram sections of a pattern were identified by Participant Two as most useful when completing the TUIC model. They commented that pattern names were the “*key to whole exercise as they were on the UI-pattern model*” and that for TUIC modelling the other sections were not relevant. Participant Two recognised that the diagrams provided guidance for the prototyping process and that the pattern model identified the relevant patterns and that the diagrams removed any possible ambiguities.

Participant Two suggested that two versions of the detailed instructions for applying the method were required. Participant Two thought Steps C and D could be amalgamated for experienced UI developers but recognised that those with less experienced may need the level of detail in the current instructions. On the other hand Participant Two considered Step E should be broken down into five smaller steps. About TUIC Modelling Participant Two was very enthusiastic making the comment “*This was the fun part and made it easy to see how the UI would come together*”. The follow-up discussion supported the written comment with Participant Two saying how really enjoyable this session had been. The advantage of first developing the UI-pattern model was also acknowledged.

The TUIC model Participant Two created from the exemplar UI-pattern model (Appendix A29) provided scored ninety eight percent, and was just missing the comment associated with the four transaction spaces to indicate they must be synchronised to the client's details for that period of occupation. Participant Two rated TUIC Modelling as being very helpful commenting that it "*bought together elements of the big picture and identified the way forward*".

Communication

Participants Two rated using patterns as extremely helpful commenting "*Technical people often use technical terms, the patterns allows unfamiliar items to non-professionals to be discussed. They can have input on what works for them*". Participant Two rated UI-pattern modelling as being very helpful in aiding communication between design team members indicating that it would help ensure that the UI met the users' real needs. Participant Two considered that using patterns and UI-pattern modelling would help non-professionals to participate more fully in the design process commenting "*levels the playing field so everyone knows how the big picture will come together*".

TUIC Modelling was rated by Participant Two as extremely helpful and similar comments to those for UI-pattern modelling were made. Participant Two saw the use of patterns and both modelling techniques as bridging "*the gap between technical and non technical members meaning level of understanding is consistent between non-professionals and professionals*".

A24.3 Participant Three

Participant Three is a professional applications developer who is currently employed to bring a research project system up to the standard required of a commercial product. This work includes the re-development of the existing user interface. Participant Three has had ten years IT experience mostly in Asia and is a non-native English speaker. Participant Three self rates as having average proficiency and as not being particularly confident saying "*My experience focussed on database applications. Interface development to me was just for completing tasks for users*". The method they currently use for UI development was described as "*Understand the user requirements; write down most important tasks the UI is expected to achieve; if problem is complex and I have time, write use cases, implement most important tasks, test, if problem is found to be complex, consider redesign the interface and user interactions*". The UI development techniques used were Use Cases and Rapid Application Development. Participant Three reported that they were not confident using lo-fi prototyping as in their experience "*I find the implementation is just totally different from the initial thoughts/plan. I find it difficult to prototype the user interface if I'm not sure of the details of the program [functionality]*".

Other than tools associated with large databases systems, the IDEs with a visual components toolbox, particularly the Delphi IDE integrated with Model Maker (CASE design tool) and PowerBuilder.

UI-Pattern Modelling

Participant Three completed the UI-pattern modelling session in two hours and twenty-two minutes. Fifty-three minutes were required for the demonstration and associated discussion which focussed on interpreting the UI-pattern model. The participant asked many questions covering different aspects of both patterns and the model. The word canonical generated some discussion. The discussion included looking at the reference section for Pattern '03 - Current Properties' along with the chain of patterns found in the example UI-pattern model starting with Pattern 03.

It took Participant Three nineteen minutes to study the requirements documentation for the exercise followed by forty-eight minutes to complete the UI-pattern modelling exercise. At the start, some time was spent matching the Use Cases with associated interaction space tables. When starting to select patterns many references to the exemplar UI-pattern model were made and then selections were checked back to the requirements document. The first attempt at creating a UI-pattern model was upside down but Participant Three self-corrected when asked to explain their model. At this point, an explanation of the patterns list and the lists of numbers under each pattern's context and reference headings was asked for. The researcher used these lists to draw part of the pattern map showing how the first few patterns were related to each other. When choosing patterns from the reference section they sketched diagrams to clarify their understanding of the patterns. Help was requested for how 'Room details' and 'Room History' might be represented and how any selected patterns would be shown linking to '07 – Collection beside Content'. While working on selecting patterns Participant Three said that "*forces are the attributes of the problem*" and this phrase was also used in answering the Exit Questionnaire.

Participant Three's opinion was that the new user of patterns would probably find the examples, context, problem and solution sections most useful but as experience is gained then the name, context and reference sections would be most useful. The Forces and Discussion sections were identified as the least useful because considering the problem and examining the examples provided enough information to be able to select or reject a pattern. The TUIC model diagrams were not considered particularly helpful. The examples were preferred. They said "*I tried to understand the solution by examining the diagram but I found the diagrams for different patterns are presented similarly so I cannot tell the difference of the diagrams instantly. But by viewing the examples I can understand instantly.*" Patterns were generally rated positively because "*The UI pattern model suggests the most proper solution for a problem. Constructing a UI pattern model might reduce later changes to the final UI.*"

UI-pattern modelling was rated as very good with the only suggestion being that the detailed instructions could indicate better where repeats occurred. They considered "*Constructing an effective UI pattern model would be useful especially helpful for a complex system with lots of tasks*". Difficulty with abstraction was acknowledged supporting the comment "*some of the middle layer patterns which might be to complete the link structure but that I think are unnecessary*".

The UI-pattern model created scored eighty-eight percent, but lacked some low-level details (Appendix A29). Neither room identification information nor a client's name would be visible when other information was accessed.

TUIC Modelling

It took Participant Three one hour and thirty-three minutes to complete the TUIC Modelling session. The introduction and demo took twenty-seven minutes. Participant Three again asked about the meaning of ‘canonical’ and ‘interaction space’ then remembered asking about these during the UI-modelling session. More detailed explanations of the three sets of symbols on the TUIC components sheet were requested. At the end of the demonstration Participant Three browsed the diagrams illustrating some of the patterns used in the exemplar and in discussion matched the exemplar TUIC model to the associated UI.

The UI requirements documentation for the exercise was browsed for five minutes. The UI-pattern model and the associated TUIC model from the demonstration were then arranged to be easily visible on the work area. The UI-pattern model exemplar provided to guide the exercise was reviewed before patterns representing the main interaction spaces were selected for level two of the UI-pattern model. They used the work area to spread the patterns identified by the UI-pattern model so that the diagrams could be seen. The patterns were occasionally turned over to check the examples. Explanations were requested for the equivalent diagrams on patterns ’12 – Choice from a Large Set’ and ‘6 - Master with Details’. When they went to redraw their prototype the researcher pointed out that position was not significant although nesting was. They used the comments on the demonstration example as templates for the comments added to their TUIC model. TUIC Modelling took twenty-six minutes.

While completing the relevant sections in the Exit Questionnaire they asked for clarification of some questions and also reviewed patterns and relevant demonstration examples before answering. The identifying Number, Name and Diagram were deemed the most useful parts of a pattern when completing the TUIC model because “*the UI-pattern model has already provided information of relationships between the different parts, I didn’t need the context or reference sections*”. Participant Three agreed that the diagrams helped them locate the CAP and navigation symbols required for their TUIC model rating the diagrams as being extremely helpful commenting “*applying the diagrams illustrating the patterns identified by the UI-pattern model, I can construct the TUIC model very easily*”.

No modifications or omissions in the TUIC Modelling method were identified by Participant Three.

In the final discussion Participant Three asked why pattern 07 was named ‘Collection beside Content’ rather than ‘List beside Content’. With reference to the examples from Visio and Google Maps illustrating pattern 07, the discussion covered the ideas of abstraction and conceptualisation.

Participant Three created a TUIC model (Appendix A29) that scored ninety-four percent. All that was missing was the detail for the table representing the room selection list along with comments that the four transaction surfaces must be synchronised to the client for the period of occupation and that only one of these surfaces would be visible at any one time. The TUIC Modelling was rated as being very helpful because “*it clarifies different problems and requirements and expresses information (actions, materials, relationships etc) clearly*”.

Communication

Using patterns to help non-professional members communicate better with other UI design team members was rated by Participant Three as only helpful commenting that “*the names of each pattern are not straight forward. They are somewhat abstract*.” But They did consider that using patterns and creating a UI-pattern model would be an extremely helpful process for teams comprised of both professionals and non-professionals commenting “*when deciding to pick a pattern, the team member has to clarify the purpose of this action and the relationships with other decisions*.” Their overall response to using patterns as gauged from responses to the Patterns Questionnaire was positive, responding negatively to only two statements. These responses confirmed that the pattern names were not always clear for this participant and information could be difficult to remember.

Participant Three considered using patterns and the UI-pattern model to guide the development of the TUIC model would be very helpful when working in a design team consisting of both professional and non-professional members. The associated comment was that “*patterns organize the complex situation or problems clearly into a whole as a UI-pattern model so that all team members can communicate based on it as they build the TUIC model*” Finally, individual patterns were considered “*easy to understand*” but a ‘*learning curve*’ to mastering the different modelling methods was recognised.

Appendix A25: Study Three – Exit Questionnaire Responses

To ensure anonymity the comments have been amalgamated. The comments have been reorganised for easier reading of associated ideas.

A25.1 UI-pattern Modelling

Justification for most useful pattern section

⇒ The name because it gave us a quick reminder of what the patterns was for (i.e. its purpose)
⇒ The context section allowed me to identify 'where to from here'
⇒ The context and reference sections because we used these for building the tree (upwards and downwards)
⇒ Images were confirmation that chosen pattern met requirements
⇒ The diagram and examples because they provided a quick way of understanding in what situations the pattern can be applied.
⇒ If a new user of the patterns the problem, context, solution and examples are the most useful. For a more experienced user name, context and reference are the most useful

Justification for least useful pattern section

⇒ The solution section required more figuring out than images [examples] did but provided essentially the same information
⇒ The forces and discussion sections because when you become familiar with the patterns you don't need to look at these anymore
⇒ Forces are the attributes of the problem, but by viewing the problem I can understand it very well without reading forces, similarly for the discussion

The UI-pattern model

⇒ The UI pattern model suggests the most proper solution for a problem.
⇒ It is a multi-dimensional map that allows and directs creation of a solution specific to requirements
⇒ Constructing an effective UI pattern model would be useful especially helpful for a complex system with lots of tasks.
⇒ It can help you identify more effective ways of designing the UI. But it is very time consuming to develop.
⇒ Constructing a UI pattern model might reduce later changes to the final UI.

Using the UI patterns when creating the UI-pattern model

⇒ An easier way of flicking between the patterns would be nice
⇒ A great way of piecing together a solution
⇒ Some way to helping you become familiar with the patterns set more efficiently and effectively at the beginning could be helpful so you can get started more quickly

⇒ UI pattern modelling can be confusing e.g. how to use some of the middle layer patterns which might be to complete the link structure but that I think are unnecessary.

The UI-pattern modelling building method

⇒ Useful to know that this is not a step-by-step process
⇒ Maybe some use of colour in the description of each step would help emphasis particular parts
⇒ The method is very good for constructing the UI pattern model

A25.2 TUIC components

Justification for most useful pattern section

⇒ We used the name to find the pattern we wanted
⇒ By looking for Identifier-name, I can find the correct pattern.
⇒ Name was key to whole exercise as they were on the UI-pattern model
⇒ Solution with the diagram showed how to present but also guided how next item should be added to the TUIC modelling.
⇒ We used the diagram in the solution it helped us visualise how we would draw the associated section of the TUIC model
⇒ The diagram identified the TUIC components needed for constructing the different parts of the TUIC model

Justification for least useful pattern section

⇒ As the UI-pattern model has already provided information of relationships between the different parts, I didn't need the context or reference sections.
⇒ Didn't need to use other parts when constructing the TUIC model (not name, solution or diagram)
⇒ The other items were useful creating the UI-pattern model but not needed in creating the TUIC model (not name, solution or diagram)
⇒ I didn't need forces and discussion as they are too detailed but I sometimes used the examples to check I understood the pattern.

The TUIC Modelling

⇒ I think the TUIC model building method should be simplified because it is quite wordy – maybe OK for a complicated and large exercise but too much for this exercise.
⇒ Maybe two sets of steps – one for experienced UI developers and another for inexperienced UI designers

Using the UI-pattern model to guide TUIC modelling

⇒ The patterns took away any ambiguity.
⇒ The UI-pattern model already developed was very relevant
⇒ I hadn't done CAP construction for a long time (more than three years), so the diagram section in the UI patterns helped me visualise how I should model the different parts of the TUIC model

- ⇒ By applying the diagrams illustrating the patterns identified by the UI-pattern model, I can construct the TUIC model very easily
- ⇒ Diagram section of UI patterns very helpful
- ⇒ This was the fun part and made it easy to see how the UI would come together

Usefulness of TUIC modelling for conceptual UI modelling

- ⇒ Clarity – all came together for me.
- ⇒ TUIC modelling allows you to more easily discuss with and explain to others your ideas for how to design the UI
- ⇒ TUIC modelling bought together elements of the big picture and identified the way forward
- ⇒ TUIC modelling is quite useful, as it clarifies different problems and requirements and expresses information (actions, materials, relationships etc) clearly

A25.3 Opinion of TUIC model diagram illustrating pattern solutions

After UI-pattern modelling

- ⇒ TUIC model diagram required more figuring out than images [examples] did but provided essentially the same information
- ⇒ I tried to understand the solution by examining the diagram but I found the diagrams for different patterns are presented similarly so I cannot tell the difference of the diagrams instantly. But by viewing the examples I can understand instantly.
- ⇒ Referring to the diagram helped me more quickly visualise the solution and decide whether the pattern was suitable or not

After TUIC modelling

- ⇒ The TUIC components illustrating the solutions guided the process
- ⇒ Because the patterns had the CAP symbols on the diagram I didn’t need to try and remember what one I should use
- ⇒ I can copy the most important symbol from the pattern (usually the upper-left symbol in the diagram)

A25.4 Communication between UI Design Team Members

Patterns

- ⇒ Team members will have a common terminology and can therefore more easily and quickly understand each what other members mean.
- ⇒ Technical people often use technical terms, the patterns allow items unfamiliar items to non-professionals to be discussed. They can have input on what works for them
- ⇒ Team members can more easily point out why a solution might be suitable or unsuitable, or more effective or not, by referring to a pattern.

⇒ Might not be very helpful as the names of each pattern are not straight forward. They are somewhat abstract.

UI-Pattern Modelling

⇒ I think UI-pattern modelling would make communication easier and faster

⇒ Communication will improve because when deciding to pick a pattern, the team member has to clarify the purpose of this action and the relationships with other decisions.

⇒ Because members of the UI design team can more efficiently modify the user interface plan, they won't have to draw and redraw paper prototypes.

⇒ Members can quickly see what the proposed interface will offer and can therefore more easily and efficiently identify things that should be modified or things that are missing.

⇒ Help to prevent the swing-type effect (a mismatch between what the user expects and the UI created)

⇒ UI-pattern modelling levels the playing field so everyone knows how the big picture is coming together

TUIC modelling

⇒ Team members can more easily argue their point of view about modelling things in the TUIC model when they have the UI patterns solution (text and diagram) sections to help support their argument

⇒ The patterns organize the complex situation or problems clearly into a whole as a UI-pattern model so that all team members can communicate based on it as they build the TUIC model

⇒ Brings clarity to the design process and means everyone is working from same base point

⇒ I think TUIC modelling would speed up the process of designing a good UI

⇒ Much easier to complete TUIC modelling using the UI-pattern model as a plan compared to when TUIC modelling from just the requirements

Assisting Non-professional Team members

⇒ Bridges the gap between technical and non technical members meaning level of understanding is consistent between non-professionals and professionals.

⇒ For non-professionals they are building an understanding of the design process and logic which will also help them understand why the final product is different to anticipated solution

⇒ Patterns are easy to understand, but to get familiar with all of them is not a simple thing. There is a learning curve to get over before using these methods.

Appendix A26: Study Three - Observations of UI-pattern Modelling Sessions

A26.1 Participant One

Activity	Time (mins)	Observations	Queries	Interventions
Intro to session	11	Subject spent a long time studying the overview, reading for meaning		
Demo	48	Watch intently while I demonstrated building the exemplar UI-pattern model	Asked for clarification of the role of the context and reference sections for selecting patterns.	
		Independently studied the exemplar		
Intro to exercise	23	Studied intro and exercise requirements	Confirmed which models were to be developed in the session and that only the UI-pattern model was to be created in this session. Confirmed the importance of the Interaction space and actions table in requirements documentation	
UI-pattern modelling	46	Selected 04 "information on a form" for modelling "staff details" chained it up to 03 "current properties"	Asked how much detail required to model "staff details" as noted exemplar only used pattern 03 and considered pattern 04 "information on a form" best Discussed pattern chains	
		Checked exemplar and made start for "Find a room" by selecting pattern 08 "cascading collections" - self corrected and selected pattern 07 "collection beside content" as top of the sub-tree	Participant affirmed they were unsure about their initial selection.	Drew subjects attention to the diagram for pattern 08 and asked how many lists involved in the "Find a Room" space
		Selected out a sub-set of patterns to represent parts of the "Find a room" sub-tree into a separate pile. Tried to then draw the linked structure but had trouble remembering what they had done in selecting the subset of patterns.	Discussed pattern 07 "collection beside content" reference list to determine best way to represent "room details" and "occupancy history". Had long discussion of the type of data that would be best represented by the three patterns references by 03 – patterns 04, 05 and 06; using the diagrams for comparisons.	Requested help as had forgotten why some patterns had been selected I went over the exemplar for find a course and referred back to some of the subjects verbalisations for selecting patterns in their pile. I copied out patterns' context lists for them so it was easier to locate common parent patterns.
		Added pattern 03 for top of transaction phase and then chained to 04. Found common parent for "Staff details", "Find room" and "Transaction phases" pattern 02 Reviewed model so far against interaction spaces and actions table by pointing to the different cells		

		After quick referral to exemplar added pattern 14 “Convenient environment actions” for the different sets of actions		
		Selected pattern 10 “Stack of working surfaces” to group the four transaction phase sub-trees.	Discussed the presentation of the four phases and asked whether copies of pattern 03 would be suitable way of presenting these Then discussed how to group these worked through context patterns 02, 07 and 10	
Questionnaire	15	Very thorough. Stopped and thought for while before answering each question set.	Asked for clarification of question 2 – least useful of sections used or the ones just not used.	
After Questions	3	Commented that modelling from requirements different from modelling from an existing UI.		
	146	2 hours 26 minutes		

A26.2 Participant Two

Activity	Time (mins)	Observations	Queries	Interventions
Intro to session	8	Quickly looked at all artefacts then returned to browse through the example requirements. Paused on interaction spaces list		I pointed out list of attributes was for completeness and this level of detail was not going to be covered in this exercise.
Demo	25	Spent time examining UI-pattern model for the course registration system then asked for explanations	Explanation why list of subjects not just part of “collection-by-content” – difference between pattern 08 “cascading collections” and pattern 07 “collection-by-content” – used examples of both patterns Explanation of how pattern 10 “stack of working surfaces” used to model current program and enrolment history. Asked for explanation of parent chain of 03 to 06 to 10 and position of student details.	
Intro to exercise	7	Quickly browsed scenario and requirements use cases	Queried the content of the 2 inventory tables – discussed link between tables and use cases	
UI-pattern modelling	42	Checked back to exemplar UI-pattern model before starting Used examples extensively to select to consider before checking the context and problem sections Used a type of bottom-up approach identifying the main interaction space and actions - then found a pattern to match: 04 – staff details 13 – list of rooms 12 – select room 06 – students study program Kept rest of set in ordered pile		
		Started drawing UI-pattern model without regard to the links defined in the context and reference sections. Selected 02 – high density and labelled it find rooms Linked patterns in a chain rather than a tree/network	Discussed use of exemplar and how to generalise from it -	I intervened Reviewed exemplar’s top structure Examined structure for list of courses and course details Examined sub-tree for List of courses
		Started to redraw UI-pattern model from 07	Subject reported that they had difficulty	Matched 07 collection beside detail to find room

		Selected 12 and 03 to represent next level down	conceptualising the grouping of UI ideas And requested help to link the patterns	I helped subject to layout patterns in hierarchy and helped subject match context and reference sections
		Added details of 13 for and then 15 for selecting room sub-tree Then repeated for room details using reference links successfully	[appear to make better choices when directed to examine diagrams]	I explained how diagram was an abstract interpretation of the examples as subject seemed to fix on just one example when looking at patterns. Discussed need to identify essential aspects of a solution rather then rushing to identify a toolbox component approach.
		Subject selected 02 as root Selected 10 to represented transaction space	I had to remind subject to check example images to confirm choices and rejections	I helped subject to focus on context section to link in staff details
			[subject decided to stop and move on as they had a two hour time constraint]	
Questionnaire	12		Checked I knew what the SWING effect was	
After Questions	3		Commented again that they found conceptual modelling and abstraction very difficult	
	97 minutes	1 hour thirty-seven minutes		

A26.3 Participant Three

Activity	Time (mins)	Observations	Queries	Interventions
Intro to session	16	Tells me has had experience with use cases	Explanation for term canonical – "Reduced to the simplest and most significant form possible without loss of generality" Explanation of overview diagram for method Explanation of how patterns could form a pattern language Explanation of links between use cases, table of actions and interaction spaces Explanation to confirm the structure of the UI-pattern language – higher and lower level patterns	I pointed out list of attributes was for completeness and this level of detail was not going to be covered in this exercise.
Demo	37	Much discussion on structure of the UI-pattern model and how it linked together After demo subject spent time reviewing the exemplar and checked the meaning of patterns 13, 10 and 06 by browsing them	Specific query on the role of pattern 14 – Convenient Environment Actions Role of pattern 03 as 'parent' to 04, 05 and 06 – particularly chain of 03 to 04 We discussed the use of 04 for adding details for modelling course details.	
Intro to exercise	19	Just read carefully – some checking of interaction table against use case		
UI-pattern modelling	48	At start of modelling made many references to the exemplar, then switched to using the list of patterns. When recording used pattern names and numbers. Pattern matched to exemplar for modelling room list structure but reviewed each pattern selected Referred back to the exercise use case and scenario before selecting pattern 12 "Choice from a large set"	Asked for confirmation of how pattern 07 "Collection beside content" might be used Asked about the problem/solution pairs describing patterns – subject reviewed the example to check explanation	
		Subject began constructing the UI-pattern model upside down		I asked subject to explain their diagram
		Subject self-corrected and	Asked for explanation of pattern list	I drew top levels of pattern language using arrows to show context and reference links complementing each other

		Selected pattern 07 “Collection beside Content” to represent Find room. Subject drew sketches of an example of tabbed spaces compared to list representing “12 Choice from a large set”		
			Much discussion on how to represent the room details and history and how to find link between selecting a room and showing it as a reserved room.	
			Asked about linking patterns up to a common parent	Showed subject how to use context links and together to find a common parent for the subtree representing “Current programme” and the “Student Details” by finding the common parent o6 then chaining back up to 03 and finally up to 02
		When subject added link adding pattern 03 in chain and linked to 07 Added in the “staff’s details”, and linked back to 02 as common parent with “Find a room”.		
		Referred back to scenario to understand details of reservation transaction Began by adding pattern 14 at bottom of sub-tree with pattern 10 “Stack of working surfaces” Commented that the forces seemed to be attributes of the problem.	Discussion of pattern 04 “Information on Form”	Subject confirmed choices with me. I copied context list for both these patterns (14 & 10) under each other so intersection is more easily identified. I found the patterns from pile so subject could review them and decide which created the context for their solution
Questionnaire	23	Referred to patterns list a couple of times when answering method questions Also examined exemplar UI-pattern model Suggested that schematic for method should be part of the questionnaire.	Queried first question didn’t seem to understand difference between sections used and those that were most useful. Asked why many arcs into pattern 15 but pattern 12 repeated in exemplar?	
After Questions	2		Discussed use of patterns in commercial settings. I referred subject to Yahoo site and also Tidwell’s book. I mentioned cooperative systems research	
	142 minutes	2 hours twenty-two minutes		

Appendix A27: Study Three – Observations of TUIC Modelling Sessions

A27.1 Participant One

Activity	Time (mins)	Observations	Queries	Interventions
Intro and demo	28	After studying the intro reading there were no requests for an explanation	After demo examined UI-pattern model exemplar and asked about modelling a table with pattern 13	Matched the part of the TUIC model exemplar represented by pattern 12 “Choice from a large set” with pattern 14 “Tabular set” with the diagrams in the patterns.
			Asked about modelling the chain of patterns representing the “Current programme” discussed substitution by the lowest pattern in chain pattern 03 by pattern 11	
			Discussed use of labels on TUIC model diagram for pattern 15 “convenient environment actions”	
			Asked about the alternative diagrams used in a number of the patterns e.g. 8 and 12	I went through the TUIC components and discussed active materials as shorthand for combinations of the materials and tools And then examined alternatives for 8 which showed embedding of pattern 07 in one alternative
Reviewed Requirements	2	Very quick browse through documents and then made the example and the UI-pattern mode prominent on the work surface	Discussed relevance of layout of the TUIC components in the abstract prototype	
Abstract prototyping	18	Started quickly to draw surrounding context frame and referred back to requirements documents to identify correct name.		
		Referred to the exemplar models as primary source of information and only referred to the patterns when unsure of details	Very brief discussion which was of the form of confirming a match between the models and TUIC model diagrams to pattern “this is – that” form Verbalising thoughts not requests for explanation	
		Subject worked from right/top to the bottom/left through the modelling exercise leaving space for adding details lower in UI-pattern model		
		Extended model to include synchronising the phases of the transaction Went back and reviewed model Made comment that could modify model for “room selection” by including more table details		

		and had not added more comments for pattern 15 “Pointer		
Questionnaires	40		For TUIC questions asked for confirmation of the role of the diagram in relationship to the solution. Asked about communication question as wasn’t sure if the implication was that the group would have a complete UI-pattern model available. Confirmed development was by a group of designers.	
Discussion	5		At end made the comment that having the UI-pattern model made TUIC modelling much easier than working directly from Use Cases as the subject had had to do for part of a class assignment about three years ago.	Note That course included a couple of lectures, a paper covering the topic and an exemplar before students set exercise to complete.
	93 mins	1 hour 33 minutes		

A27.2 Participant Two

Activity	Time (mins)	Observations	Queries	Interventions
Intro and demo	18	Examined UI-pattern model for example Watched and listened without interruption to demo other than confirm understood and I should proceed	Participant made an aside comment that they preferred to learn by doing.	
Reviewed Requirements	7	Browsed requirements		I spent some time talking through UI-pattern model and related interaction spaces table
Abstract prototyping	10	Quickly started by drawing an enclosing context and named	Asked for clarification of requirements to logon	We reviewed prerequisites for the use case "Register hotel room"
	32	Examined exemplar and then equated "Staff details" with "Student details" Added TUIC components labelled specifically not generally	Confirmed match Asked for clarification of level of detail required for model Asked if layout and size was significant	Showed exemplars at 3 different levels of detail and indicated required level
			Asked if could tick nodes off on given UI-pattern model	Encouraged
		Had difficulty interpreting how to model pattern 13 "Choice from a large set" Rubbed out first attempt and sort help	Discussed exemplar representation of "List of courses using patterns 12 and 13 Compared two patterns (11 & 12) and discussed	Showed how to use chevron to represent multiple rows in a table
		Subject proceeded to add all sets of environmental actions to diagram using specific naming Paused for some time apparently examining the exemplar models and "pondering" my explanation	Queried how to model sub-tree for selected room – pattern chain 03-to-06 Explained that when encountering abstraction chain like this normally the lower pattern in the chain substitutes for the higher ones – compared the requisite TUIC model diagrams illustrating these patterns.	
		Completed modelling the "Find room" sub-tree using many genetic TUIC components but using specific naming.	Queried the diagram alternatives on pattern 10 - explained	.
		Started drawing context for "transaction phases" Kept patterns in order		Intervened when observed subject trying to model pattern 10 "Stack of Working Surfaces" from memory found pattern 10 and suggested they checked it out first
		Completed transaction phase modelling Once considered model complete did not review and did not complete ticking off nodes on the	Noted that when subject asked questions used pattern names and correct TUIC terminology	Prompted subject not to go into too much detail and they could consider adding comments and labels to complete diagram

		UI-pattern model		
Questionnaires	22			
Discussion	5		Subject enthusiastic about TUIC modelling said they really enjoyed doing this model and also could now see advantage of the UI-pattern modelling	
	94 mins	One hour thirty-four minutes		

A27.3 Participant Three

Activity	Time (mins)	Observations	Queries	Interventions
Intro and demo	27	Read introductory section carefully Re-examined patterns and TUIC component list	Asked about meaning of CAP and the word canonical in particular. Asked what an "interaction space" was – remember on being given an explanation. Asked for explanation of TUIC component groupings particularly the meaning of materials.	Together matched symbol list to the exemplar abstract prototype.
		After demo re-examined TUIC model diagrams illustrating pattern solutions.		Worked through meaning of pattern 4 "Information on a form" for modelling "Student details" and compared level required with detailed model in appendix. Spent some time matching detailed Abstract model with detailed required and the lo-fi prototype.
Reviewed Requirements	5	Quickly browsed document		
Abstract prototyping	17	At beginning of exercise arranged example UI-pattern model and abstract prototype together so they were easily visible. Quickly reviewed UI-pattern model for exercise and started selecting patterns for top 2 levels		I located patterns asked for from the pattern pile.
		Used tabletop to spread out required patterns so could see diagrams with minimum overlapping. On three occasions turned pattern over to check the example image	Asked for interpretation of the active materials symbols on TUIC components list. Asked for explanation for equivalent TUIC model diagrams on patterns 12 and 6.	I continued finding patterns as asked. Interjected when subject going to redraw to include stacked surfaces (10) at side and pointed out that position is not significant and below the main interaction space they had already completed was perfectly valid. Reminded that nesting was important.
	9	Subject decided enough detail added and began adding comments to diagram. used exemplar rather than patterns to decide on wording	Queried how "search for room" results and "transaction space" linked then added additional comment to clarify.	
Questionnaires	31	Spent a lot of time pondering the communications questions before answering – reviewed the patterns and exemplar abstract prototype again before answering	Asked about meaning of first question – used versus most useful Asked for name of the "first" model created – UI-pattern model. Asked for explanation of statement 11 phrase "I felt in command of …"	

Discussion	4	Reported that they found some pattern names very confusing – couldn’t see why used the term collection rather than list for pattern 07. And thought should use tabs rather than stacked surfaces, pattern 10	Didn’t seem to be able to abstract away from the one implementation they preferred even when alternative examples from patterns where presented as also being examples of a collection or stacking.	
	93	One hour and thirty-three minutes		

Appendix A28: Study Three – UI-pattern Models

A28.1 Participant One

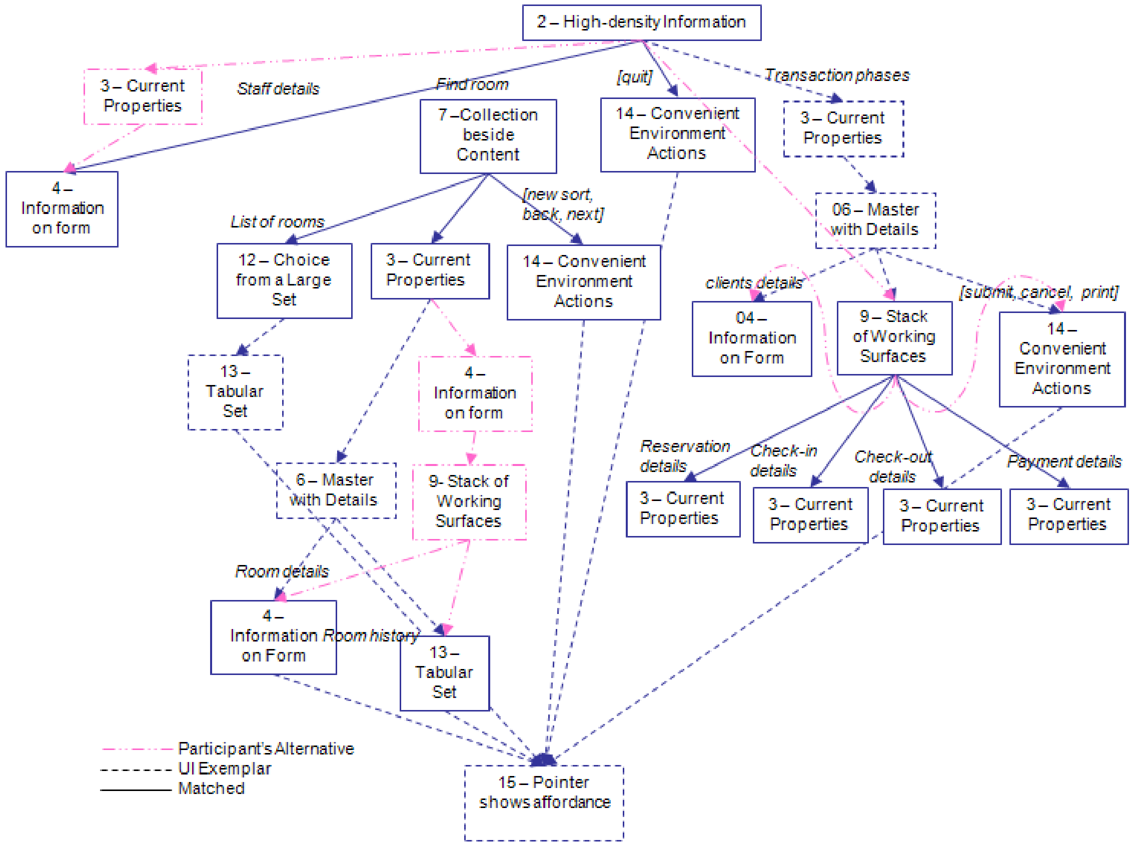


Figure A28.1 - Participant One's UI-pattern model overlying the exemplar UI-pattern model

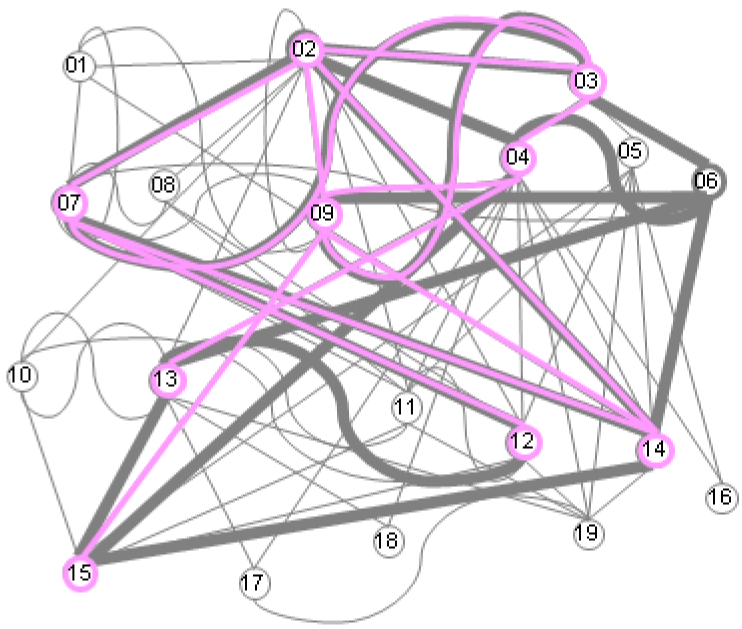


Figure 1 - Participant One's selected patterns and links overlaying the pattern map

A28.2 Participant Two

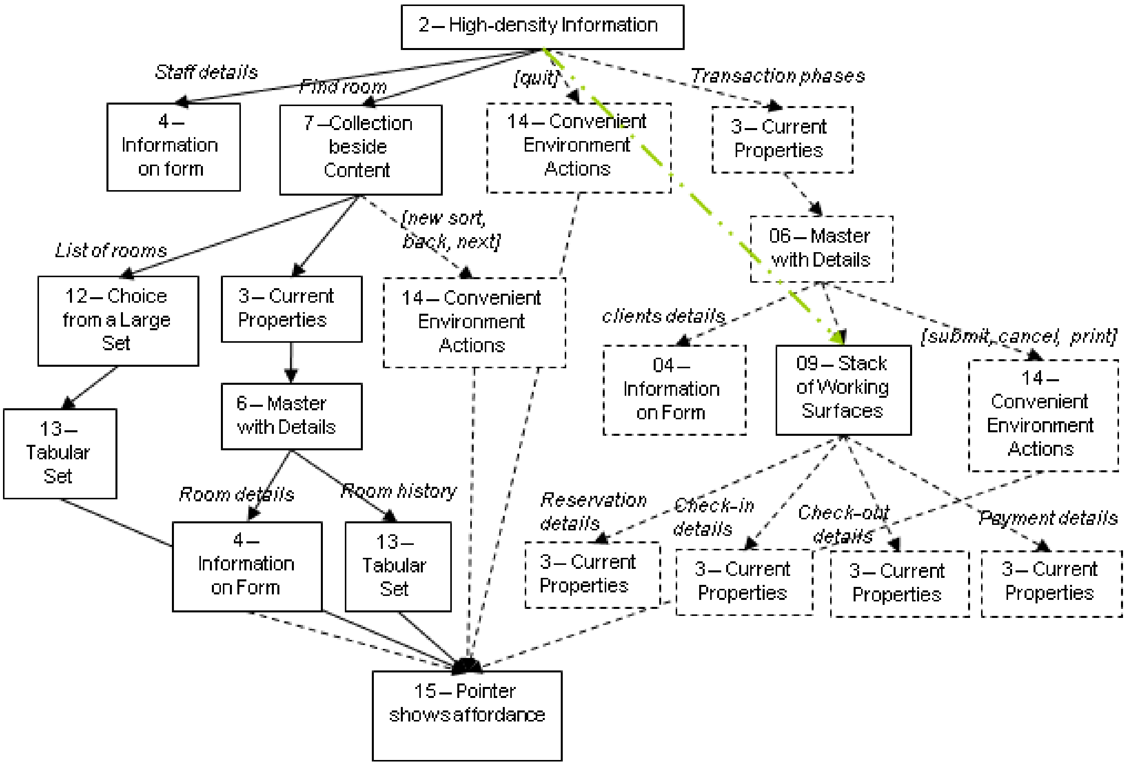


Figure 2 – Participant Two's UI-pattern model overlying the exemplar UI-pattern model

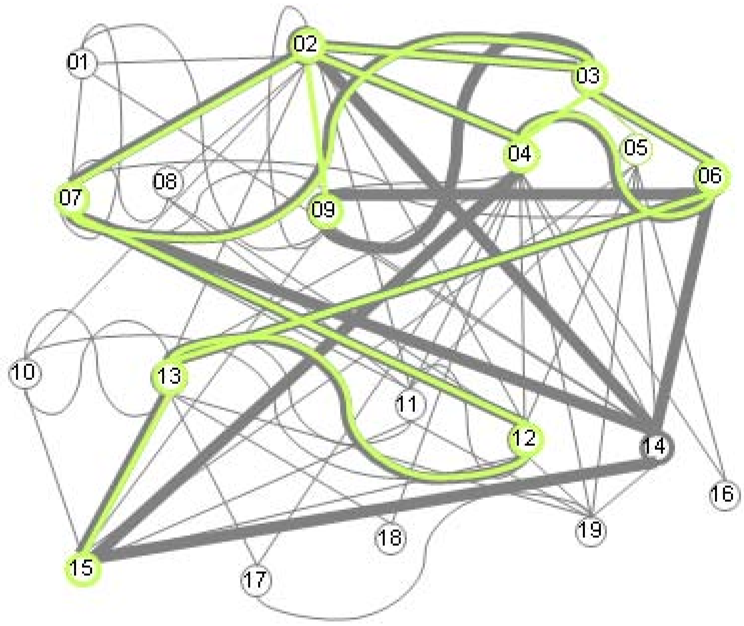


Figure 3 - Participant Two's selected patterns and links overlaying the pattern map

A28.3 Participant Three

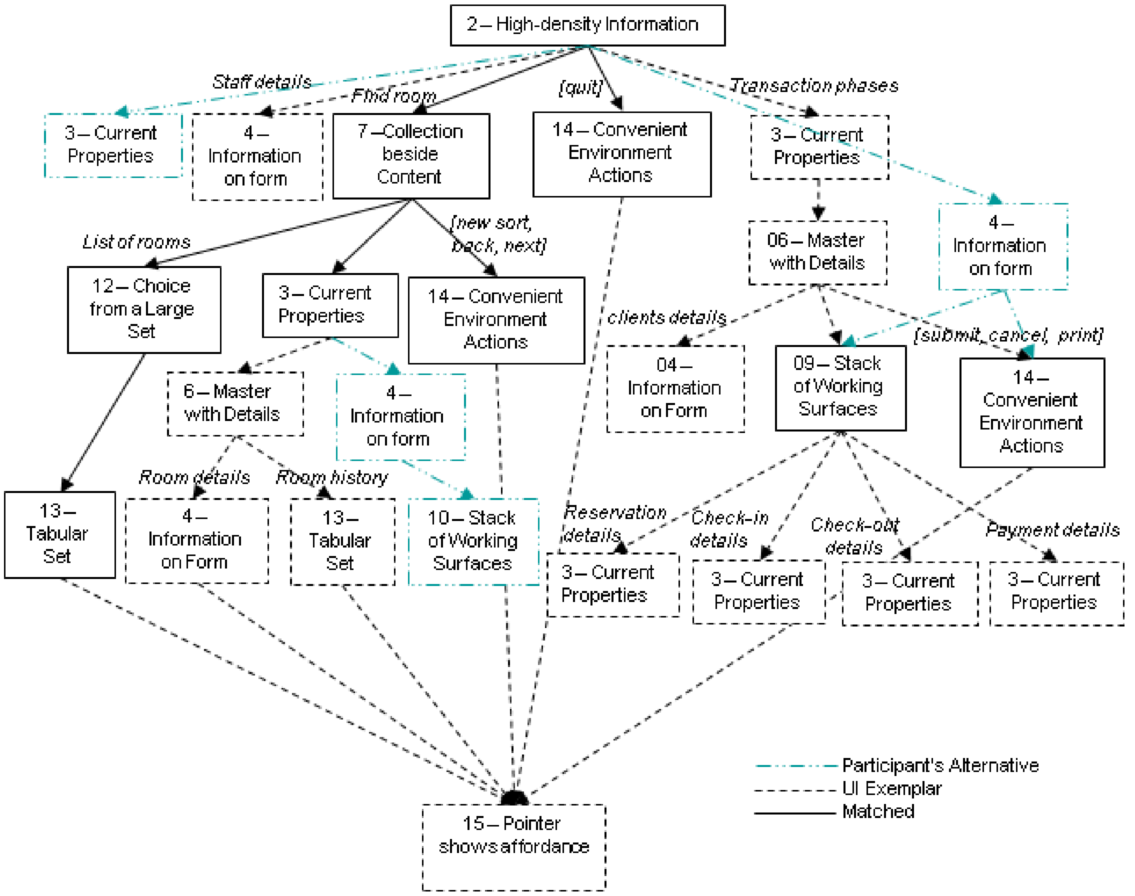


Figure 4 – Participant Three's UI-pattern model overlying the exemplar UI-pattern model

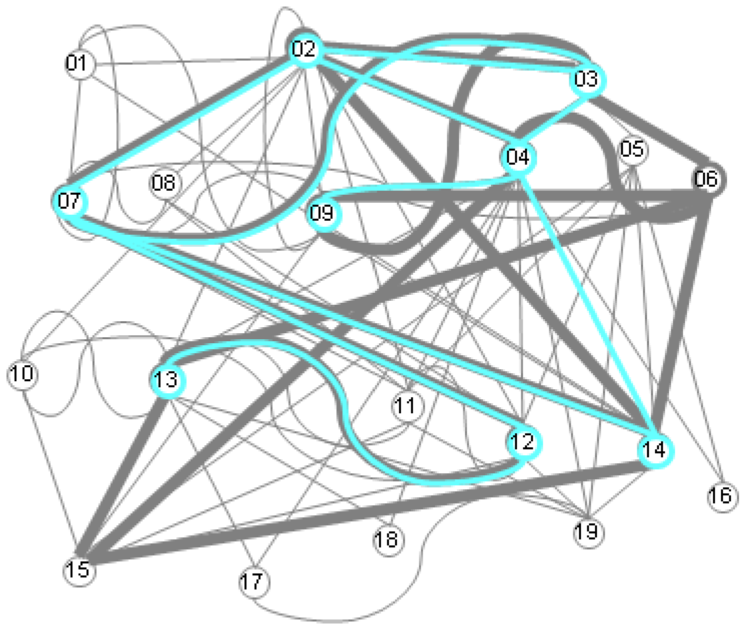


Figure 5 - Participant Three's selected patterns and links overlaying the pattern map

Appendix A29: Study Three – TUIC Models

A29.1 Participant One

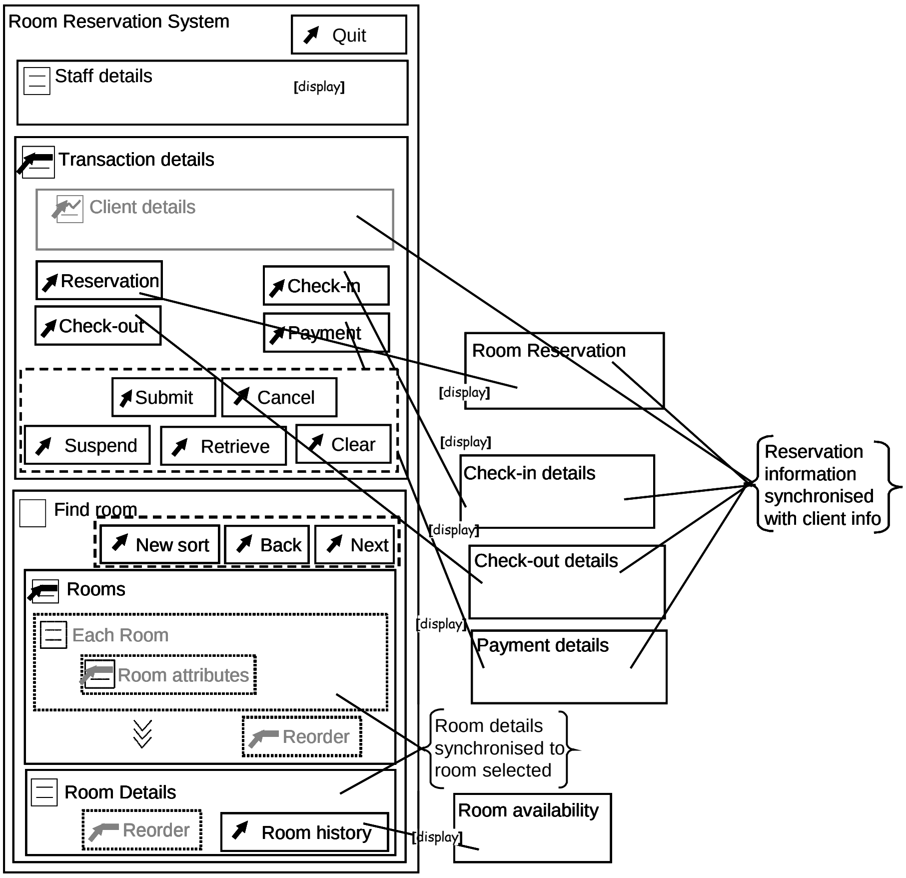


Figure 6 – Participant One's TUIC model overlying the exemplar

A29.2 Participant Two

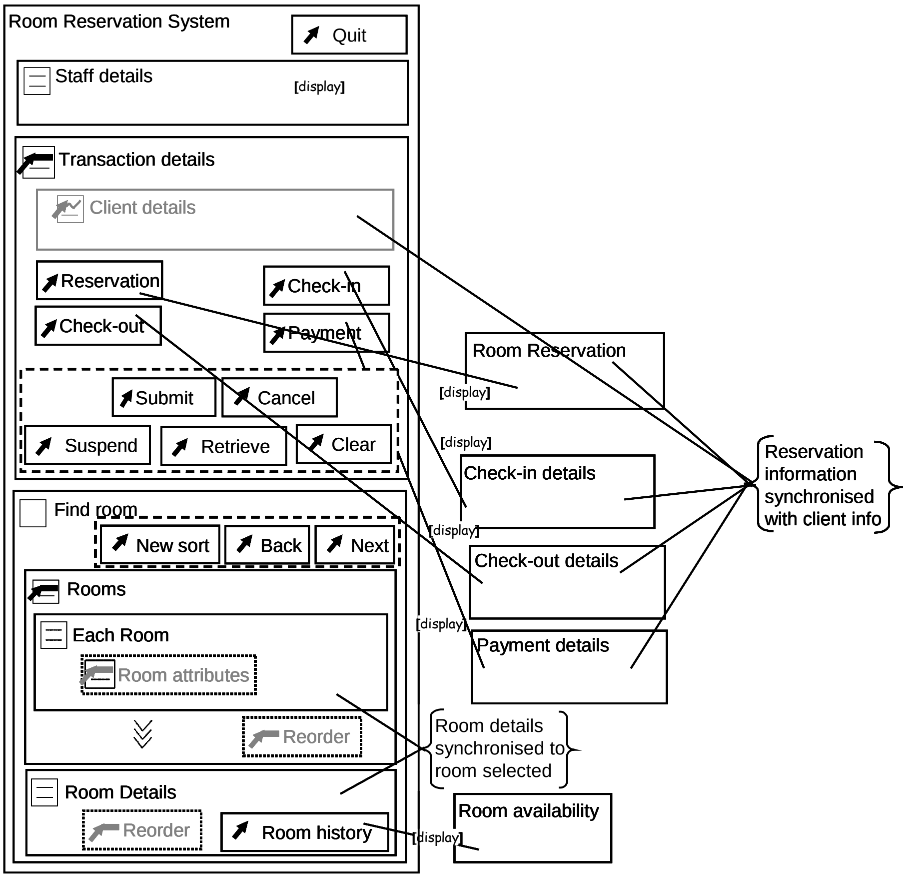


Figure 7 – Participant Two's TUIC model overlying the exemplar

A29.3 Participant Three

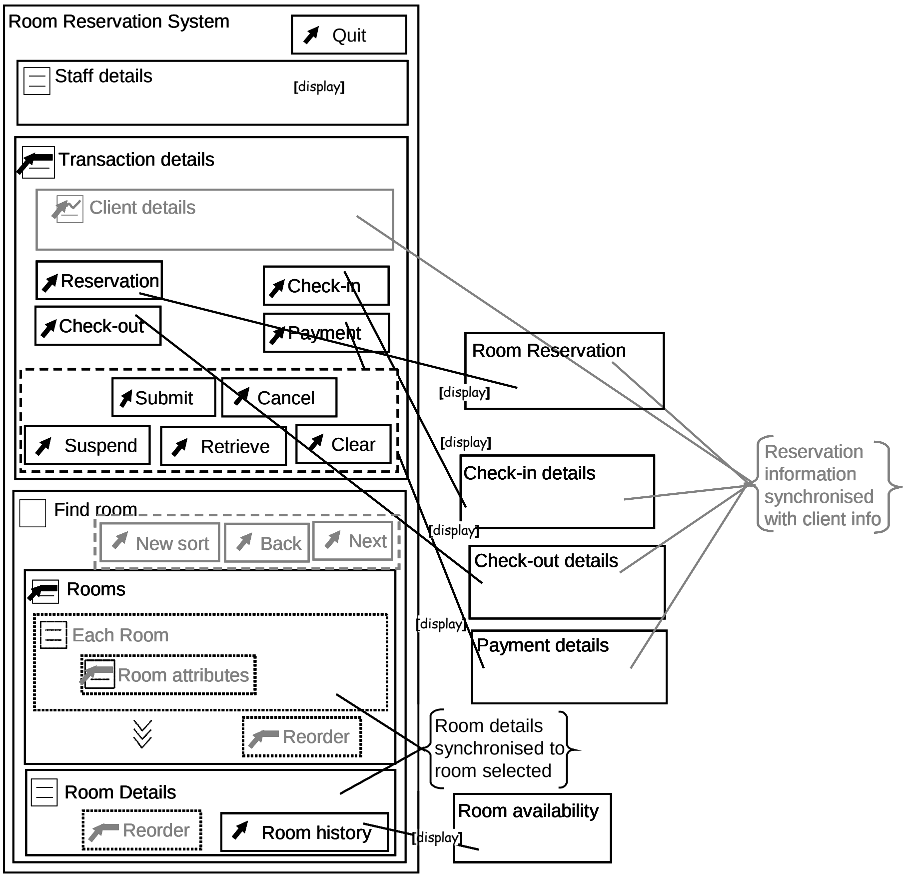


Figure 8 – Participant Three's TUIC model overlying the exemplar

Appendix A30: Study Three - Exercise Protocol

Name: ______________________________ **Ref #** __________

Preparation

Introduction to the research	1 mins ________
Participant asked to study information sheet	10 mins _______
If participant agrees to take part in the research then:	
*Complete ethics form	2 mins ________
*Complete Background questionnaire	5 mins ________
*Give pattern set to review	
	(18 mins)

Date & time: __________

Place: ______

Introduction

Give subject a copy of the Introductory manual and discuss content	3 mins ________
*Study introduction & Initial requirements	25 mins _______
*Demonstrate creation of the UI-pattern model	15 mins _______
Discussion to clarify proposed methods and worked example	5 mins ________

UI-pattern modelling

Introduction to the design exercise	3 mins ________
*Study the Hotel Reservation Requirements	30 mins _____
*Subject creates UI-pattern model	40 mins _____
*Complete UI-pattern modelling questionnaire	10 mins _______
Ask for explanation of observed behaviours	5 mins ________
*Participant given study material, patterns and CAP paper to read	
	(136 mins – 2hrs 16mins)

Date & time: __________

Place: ______

TUIC modelling

*Review example requirements and example UI-pattern model	10 mins _______
* Demonstrate creation of an TUIC model	15 mins _______
Discussion to clarify proposed methods and worked example	5 mins ________
*Review requirements and introduce example UI-pattern model	10 mins _______
*Subject creates TUIC model	40 mins _______
*Complete TUIC modelling questionnaire	10 mins _______
*Complete Pattern questionnaire	5 mins ________
Ask for explanation of observed behaviours	10 mins _______
	(105 mins – 1hr 45mins)

Thank participant

Lists of required artefacts

Preparation

- Information sheet
- Ethics approval form
- Background questionnaire
- Patterns plus list and TUIC components
- Copies of all documents so potential participants can examine them on request

Introduction

- Introductory manual (part 1)
- Patterns plus list and TUIC components
- CAP paper

UI-pattern modelling

- Hotel Reservation requirements for the modelling exercise.
- Set of UI Patterns with the list of the nineteen pattern names summarising context and reference lists.
- An identification sheet of canonical TUIC model components used to illustrate the solutions in the patterns.
- Paper, pencils and rubber for modelling.
- UI-pattern modelling questionnaire.

TUIC modelling

- Introductory manual (part 2)
- UI-pattern model exemplar
- Exercise.
- Patterns.
- Paper, pencils and rubber for modelling.
- TUIC modelling questionnaire.
- Patterns questionnaire

For Researcher – use reference number to identify observations

- Introductory manual – to annotate
- Exercise – to annotate
- Pattern list.
- TUIC components
- Paper, pencils and rubber for recording observations.
- Protocol