

# **IEC 61131-3-Based Control of a Reconfigurable Manufacturing Subsystem**

by

Albert Jakobus Hoffman

Thesis presented in fulfilment of the requirements for the degree of Master of  
Engineering (Mechatronic) in the Faculty of Engineering at Stellenbosch



Supervisor: Prof AH Basson

December 2014

## **Declaration**

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2014/05/15

## **Abstract**

### **IEC 61131-3-Based Control of a Reconfigurable Manufacturing Subsystem**

The South African industry has an increasing need for manufacturing automation. However, the classical form of automation is not cost effective for the low volumes and high variance of products that are produced there. The industry may use the reconfigurable manufacturing system (RMS) concept to improve production of its products. However, industry has been unwilling to adopt the reconfigurable manufacturing systems developed in recent research projects. Due to industry's hesitance to adopt the control platforms on which reconfigurable manufacturing systems are currently based, the focus of the thesis is on creating a reconfigurable control system using industry accepted technologies.

This research focused on evaluating a Beckhoff embedded PC's suitability as a station controller that controls a reconfigurable subsystem in an RMS. The control system for the station controller was developed using only the IEC 61131-3 programming languages and the Beckhoff programming software. This control system was evaluated by using it to control a station that is responsible for testing a circuit breaker's tripping current and time.

The developed control system was based on the ADACOR architecture because of its optimisation capabilities that were necessary to keep the cycle time of the station as low as possible. The design and implementation of the physical configuration and control system of the station is described in this thesis. The station was designed to meet the requirements of both an RMS and the case study.

Because of the limitations of the IEC 61131-3 programming languages, dynamic instantiation of holons is not possible and a method was developed to simulate dynamic task holons. By making use of the embedded PC's ability to run multiple PLCs at the same time, each type of holon was run in its own PLC thread.

The developed control system and station was evaluated by conducting experiments using a laboratory test setup. The evaluation of the developed control system in this thesis proved that an RMS can be created, in the context of station control, using IEC 61131-3 and industry accepted technologies, if a hardware platform is used that allows multiple PLCs to be run in individual threads. The control approach that was created in this thesis can be used to create station control systems that offers optimised cycle times, the benefits of an RMS and the benefits of industry accepted technology.

## Uittreksel

### **IEC 61131-3-gebaseerde Beheer van 'n Herkonfigureerbare Vervaardiging-Substelsel**

Die Suid-Afrikaanse bedryf het 'n toenemende behoefte aan geoutomatiseerde vervaardiging. Die klassieke vorm van outomatisasie is egter nie koste effektief vir die lae volumes en hoë variansie van produkte wat in Suid Afrika geproduseer word nie. Die bedryf kan moontlik die konsep van 'n herkonfigureerbare vervaardigingstelsel (HVS) gebruik om vervaardiging te outomatiseer. Die bedryf is egter nie bereid om die herkonfigureerbare vervaardigingstelsels wat in onlangse navorsingsprojekte ontwikkel is, te aanvaar nie. As gevolg van die bedryf se huiwering om die beheerplatforms waarop herkonfigureerbare vervaardigingstelsels tans gebaseer word, te aanvaar, is die fokus van die tesis om industrie-aanvaarde tegnologie te gebruik om 'n herkonfigureerbare beheerstelsel te skep.

Hierdie navorsing fokus op die evaluering van 'n "Beckhoff embedded PC" se geskiktheid as 'n stasiebeheerder van 'n herkonfigureerbare substelsel in 'n HVS. Die beheerstelsel vir die stasie beheerder is ontwikkel deur slegs van die IEC 61131-3 programmeringstale en die Beckhoff programmering-sagteware gebruik te maak. Hierdie beheerstelsel is geëvalueer deur dit op die beheer van 'n stasie wat verantwoordelik is vir die toets stroombrekers, toe te pas.

Die beheerstelsel was gebaseer op die ADACOR argitektuur as gevolg van die optimeringsvermoëns wat noodsaaklik was om die siklustyd van die stasie so laag as moontlik te hou. Die ontwerp en implementering van die fisiese konfigurasie en beheerstelsel van die stasie word in hierdie tesis beskryf. Die stasie was ontwerp om aan die vereistes van beide 'n HVS en die gevallestudie te voldoen.

As gevolg van die beperkings van die IEC 61131-3 programmeringstale, is dinamiese instansiëring van holons nie moontlik nie, en 'n metode is ontwikkelom dinamiese taakholons na te boots. Deur gebruik te maak van die "embedded PC" se vermoë om meervoudige PLCs terselfdetyd te hanteer, kan elke holon tipe in sy eie "thread" loop.

Die ontwikkelde stelsel en die stasie is geëvalueer in 'n laboratorium deur middel van eksperimente. Die evaluering van die beheerstelsel in hierdie tesis bewys dat 'n HVS geskep kan word, in die konteks van 'n stasiebeheerder, deur IEC 61131-3 en tegnologie wat wyd in die industrie aanvaar word, te gebruik mits die hardware-platform wat gebruik word toelaat dat verskeie PLCs terselfde tyd op een beheerder kan loop. Die beheerbenadering wat geskep is in hierdie tesis kan gebruik word om stasie- beheerstelsels te skep wat optimale siklus tye, die voordele van 'n HVS en die voordele van industrie-aanvaarde tegnologie bied.

## **Acknowledgements**

I would like to express my deepest appreciation to all the people who supported me in doing this thesis. I would like to specially thank the following people for their support:

- Prof. Basson, for providing me with the opportunity to do my MSc.Eng. as well as the guidance to successfully complete it.
- My colleagues, Karel Kruger and Reynaldo Rodriguez for your advice and assistance.
- Beckhoff South Africa, for your support and technical assistance.
- CBI low voltage, for case study information.
- My friends, for the continual support and understanding you gave me.
- My family, for always believing in me. Without your support and motivation I would not have been able to complete this thesis.

Above all I want to thank my father in heaven, for giving me the strength and wisdom I needed to complete this thesis.

# Table of Contents

List of Figures .....	viii
List of Tables .....	ix
List of Abbreviations .....	x
1 Introduction.....	1
1.1 Background .....	1
1.2 Objectives.....	2
1.3 Motivation .....	2
2 Literature Review .....	4
2.1 Traditional Control Architectures .....	4
2.1.1 Hierarchical.....	4
2.1.2 Heterarchical .....	5
2.1.3 Centralized .....	5
2.2 Holonic.....	6
2.2.1 Definition .....	6
2.2.2 PROSA.....	7
2.2.3 ADACOR.....	8
2.2.4 Ants.....	8
2.2.5 Example .....	9
2.3 Reconfigurable Manufacturing Systems .....	9
2.4 Control Software Implementation.....	10
2.4.1 IEC 61131-3.....	10
2.4.2 IEC 61499 .....	11
2.4.3 Agent Based Control.....	11
2.5 Evaluation of Control Architecture and Platforms.....	12
2.5.1 IEC61131-3.....	12
2.5.2 3 <sup>rd</sup> Party extensions to IEC61131-3 .....	12
2.5.3 Embedded PC .....	13
2.5.4 Agent Platforms .....	13

3	Case study .....	15
3.1	Reconfigurable Quality Assurance Cell Overview .....	15
3.1.1	Product Description .....	15
3.1.2	Cell Design Requirements .....	17
3.1.3	Physical Cell Architecture .....	18
3.1.4	Cell Control Architecture.....	20
3.2	Electronic Test Station Overview .....	20
3.2.1	Station Design Requirements (Performance and Functional).....	20
3.2.2	Station Hardware Architecture .....	21
3.3	Electronic Test Station Subsystems .....	22
3.3.1	Ramp Wave Tester.....	22
3.3.2	Robot.....	24
3.3.3	Gripper .....	25
3.3.4	Pallets and Transport System.....	27
3.4	Reconfigurability Considerations.....	29
3.4.1	Customization .....	29
3.4.2	Convertibility .....	29
3.4.3	Scalability .....	29
3.4.4	Modularity .....	29
3.4.5	Integrability.....	29
3.4.6	Diagnosability .....	29
4	Control Development.....	30
4.1	Control Hardware.....	30
4.2	Control Architecture.....	30
4.3	Data Shared Between Holons.....	33
4.3.1	Terminology in a Workspace.....	33
4.3.2	Workspaces Table.....	34
4.3.3	Work in Progress Table .....	36
4.3.4	Parity and Priority Table.....	36
4.4	Interholon Communication .....	39
4.4.1	Inbox and Outbox Buffers .....	39

4.4.2	XML Formatting.....	40
4.5	Supervisor Holon Implementation .....	41
4.5.1	Overview .....	41
4.5.2	Pairing of Task Holons .....	43
4.5.3	Priority Assignment .....	43
4.5.4	Determining the Best Position to Move to.....	43
4.5.5	Pallet Holon Functionality .....	44
4.6	Task Holon Manager.....	44
4.7	Robot and Gripper Holon.....	47
4.7.1	Robot Functionality .....	47
4.7.2	Gripper Functionality.....	49
4.8	Product Holon .....	50
4.9	Tester Holon.....	50
5	Assessment.....	51
5.1	Change within Product Family.....	52
5.2	Introduction of a New Product .....	52
5.3	Change in Origin .....	53
5.4	Change in Physical Configuration .....	54
5.5	Change in Pallet .....	55
5.6	Change in Gripper .....	56
5.7	Manual Override .....	56
5.8	Robustness.....	56
5.9	Cycle Time .....	57
5.10	Control Evaluation.....	59
6	Conclusions and Recommendations .....	61
7	References.....	63



# List of Figures

Figure 1: Hierarchical architecture .....	4
Figure 2: Heterarchical architecture .....	5
Figure 3: Centralized architecture.....	5
Figure 4: Basic building blocks of a HMS (Van Brussel, et al., 1998) .....	7
Figure 5: Exploded view of Q-frame circuit breaker.....	16
Figure 6: Product variety .....	16
Figure 7: Layout of RQA cell .....	19
Figure 8: Layout of the ETS .....	22
Figure 9: Eight position test rack mock up .....	23
Figure 10: Diagram indicating the different axis of the robot (Eurobots, 2012) ...	25
Figure 11: Picture of the designed gripper.....	27
Figure 12: Overhead view of parallel conveyor and pallet positions .....	28
Figure 13: Control architecture of the ETS .....	32
Figure 14: Terminology in a workspace .....	33
Figure 15: Connection between outbox FIFO and inbox FIFO.....	40
Figure 16: Flow chart of supervisor holon.....	42
Figure 17: Flow chart of task holon manager .....	46
Figure 18: Flow diagram of the robots movements .....	47
Figure 19: Lab setup of ETS.....	52

# List of Tables

Table 1: Example Workspace Table .....	35
Table 2: WIP table with example entries.....	37
Table 3: Parity and priority table with example entries.....	38
Table 4: RMS characteristics tested by each assessment .....	51
Table 5: Cycle time breakdown of first experiment .....	58
Table 6: Cycle time breakdown of second experiment.....	58

# List of Abbreviations

ETS	-	Electronic Test Station
FIFO	-	First In First Out
HMI	-	Human Machine Interface
PLC	-	Programmable Logic Controller
PPT	-	Parity and Priority Table
RMS	-	Reconfigurable Manufacturing System
RQA	-	Reconfigurable Quality Assurance
RWT	-	Ramp Wave Tester
WIP	-	Work In Progress

# 1 Introduction

## 1.1 Background

Industries today increasingly require manufacturing lines to be reconfigurable and adaptable to changes, but the manufacturing systems that are commonly being used in factories are fixed in purpose or product range. In South Africa, many industries compete in niche markets which lead to multiple product changes in a short time, and this poses particular challenges to automate.

Quality assurance is a huge concern for the modern industry, especially when mission critical devices are being manufactured. Quality assurance is introduced into a product's manufacturing line to ensure that the product meets or exceeds the design specifications and that the products are correctly labelled. It is therefore important to have a quality assurance process that is capable of conducting tests which yield repeatable and consistent results. Quality assurance checks are commonly done by humans in South Africa and this leaves much to be desired in terms of consistency and repeatability. Replacing a manual quality assurance station with a fully automated reconfigurable machine would not only ensure consistency and repeatability, but it could also decrease production time and provide traceability of the products. This also reduces scrap produced and, more importantly, it reduces the risk of failures in the market.

A common concern for companies looking to automate their production lines is possible disturbances in production, like breakdowns. All systems have breakdowns or sensor failures at some point and this causes production to halt until the problem is resolved.

These product variety, quality and disturbance considerations create the need for an automated testing and labelling system that is adaptable to product changes and disturbances. Creating a reconfigurable manufacturing system (RMS) is a promising solution to this problem. Although research has been done in creating systems that are reconfigurable, these systems were limited to test cases only, since industry has not yet adopted the developed technology.

The RMS concept has been investigated at the University of Stellenbosch since 2008 (Sequeira, 2009) (Dymond, 2009) (Adams, 2010) (Le Roux, 2013) (Mulubika, 2013) (Kruger, 2013). The case study for their research was the manufacture of the subassembly of a circuit breaker. Agent Based Control (ABC) and IEC 61499 were used to create the RMS. Although an RMS was successfully created, both IEC 61499 and ABC failed to be accepted by industry. This agrees with other researcher's findings regarding the acceptance of ABC in the industry (Marik, 2005) (Leitão, 2009).

CBI Electric: Low Voltage, the South African industry partner supporting this research, wants to improve its production efficiency to gain competitiveness within its market sector. The company assembles circuit breakers in a factory in Lesotho. Due to the nature of its products, quality assurance is of utmost importance and therefore the company wants its quality assurance process to be consistent, traceable and repeatable. CBI currently uses manual labour to do the quality assurance and this fails to meet the above mentioned criteria. The company is considering implementing automated cells that are capable of testing all of its current and future products.

A concept for a quality assurance cell will be developed for CBI by the Mechatronics Automation and Design Research Group at the Stellenbosch University. The quality assurance cell will be used as a case study for the research group with different subsystems being assigned to different members of the group. The main focus of the group is developing an RMS using industry accepted technologies so that the industry will be more likely to adopt the RMS concept.

The author of this thesis was assigned the Electronic Testing Station and Beckhoff equipment for the control of the station. Rainer Graefe and Darlington Masendeke were assigned to the Riveting and Stacking Station and will each use Siemens and National Instrument's Labview, respectively, to control the station. The conveyor and cell controller were assigned to Karel Kruger and Marcus Kotze. The industry accepted technology that they would use for the controllers has not been finalized at the time of writing this thesis.

## **1.2 Objectives**

This thesis will focus on evaluating a Beckhoff embedded PC's suitability as a station controller that will control a reconfigurable subsystem in an RMS. This developed control system will be evaluated by using it to control one of the subsystems in the CBI case study, i.e. the station that is responsible for testing the breaker's tripping current and time using test machines developed by CBI. Evaluating the new control concept for such a reconfigurable system should provide sufficient evidence to prove the application of the technology to the automation sector.

## **1.3 Motivation**

Over the past decade there have been numerous papers published about reconfigurable manufacturing systems and, although most include working concept demonstrators, no proof has been found up to this point that the technology has been applied in the industry (Leitão, 2009) (Marik, 2005). A possible reason that the industry has not yet adopted this technology is the cost involved. Normally a company would have a manufacturing line built for a specific product and when the product is no longer produced, the line is scrapped.

Manufacturing products in this manner is only profitable if large quantities are produced (Koren & Shpitalni, 2010). Because the South African industry competes in niche markets, low volumes and high variances in products are normal. This means that the South African industry is not suited for traditional automated lines when considering the low production volumes. Therefore, there is a need for a manufacturing system that is flexible and reconfigurable.

Industry commonly relies on industrial embedded controllers, such as programmable logic controllers (PLC), to control the machines in the production lines. However, RMSs developed in research institutions commonly used other, more capable, but less robust, controllers implementing agent based control. It would improve the chances of industry adopting the RMS concept if its control is based on industry accepted controllers like PLCs. To date researchers have considered PLCs to be unsuitable for RMS controllers. This thesis will reconsider this position, particularly in the context of a station controller in an RMS.

By using an embedded PC for the station's controller, the programmer of the station would have access to features and capabilities that is not available to PLCs. An RMS that uses an embedded PC also has a greater chance of being accepted by the industry as embedded PCs are already accepted by the industry.

## 2 Literature Review

This chapter first describes the traditional control architectures that are commonly used in industry. These architectures are discussed as they form the basis of other more advanced architectures discussed later. Holonic control is then defined and the most popular holonic manufacturing systems are described. The holonic control concept is closely related to RMS. The RMS concept and its characteristics are described in the third section. The approaches used to implement the architectures are discussed followed by an evaluation of the control architecture and platforms.

### 2.1 Traditional Control Architectures

The following subsections describe the control architectures commonly used in industry with PLCs, summarising the layout of the controllers in the different architectures, as well as the advantages and disadvantages of the architectures.

#### 2.1.1 Hierarchical

In this architecture there is only one main controller that has control/authority over its sub-controllers (Figure 1). Each sub-controller has control/authority over its sub-controllers. Information moves from the bottom of the structure to the top and commands move from the top to the bottom. The advantage of this type of control is that it is good at global optimization since the main controller has knowledge of all subsystems. The disadvantage of the hierarchical control architecture lies in the rigidity of the system, which implies a weak response to disturbances in the system (Leitão & Restivo, 2006). If a fault or error occurred in a sub-system, the whole system would halt and cease production until the issue is resolved.

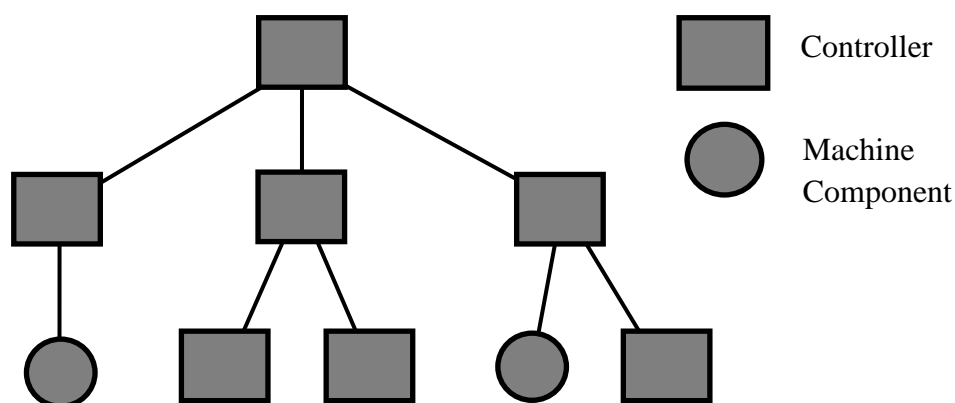
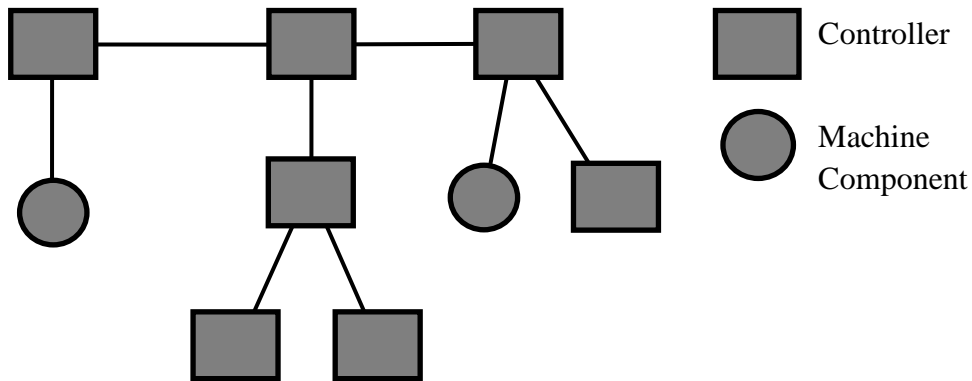


Figure 1: Hierarchical architecture

### 2.1.2 Heterarchical

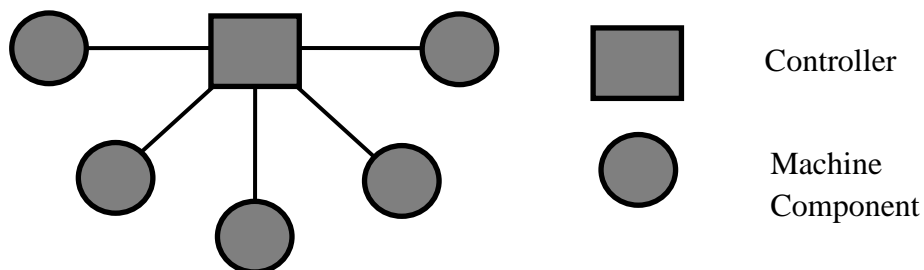
In the heterarchical control architecture there is more than one main controller and all are on the same level in terms of authority (Figure 2). Each main controller may have several sub controllers working for it. This architecture responds well to disturbances, but because each controller only has partial knowledge of the system, global optimization cannot be guaranteed (Leitão & Restivo, 2006).



**Figure 2: Heterarchical architecture**

### 2.1.3 Centralized

The centralized control architecture relies on one main controller that controls the entire system (Figure 3). This requires the main controller to have knowledge of all the systems connected to it. This control architecture is very similar to the hierarchical architecture in that one controller is in charge of the system. The key difference is that, instead of issuing commands that get passed down the hierarchy to controllers of the relevant subsystems, the centralized controller controls the subsystems directly. This approach shares the same advantages and disadvantages as the hierarchical approach, but it is also more complex to change.



**Figure 3: Centralized architecture**



## 2.2 Holonic

### 2.2.1 Definition

The word holon is derived from the Greek word *holos* which means whole, and the suffix *-on* which, as a proton or neutron, suggests a particle or part (Koestler, 1967). Van Brussel et al. (1998) defined holons to be simultaneously self-contained wholes to their subordinated parts and dependent parts when viewed from the inverse direction. Van Brussel et al. (1998) then used the holon concept to develop the holonic manufacturing paradigm. They also noted that the “holonic architecture shall enable easy configuration, easy extension and modification of the system and allow more flexibility and a larger decision space for higher control levels”, which corresponds well to some properties of RMSs.

In industry a holon is defined as an autonomous and co-operative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects (Van Brussel et al., 1998). A holon consists of an informational processing part and often a physical processing part. A system of holons co-operating to achieve a goal or objective is defined as a holarchy. Holons makes use of the distributed control structure, but are governed by basic rules for co-operation that are defined by the holarchy (Van Brussel et al., 1998).

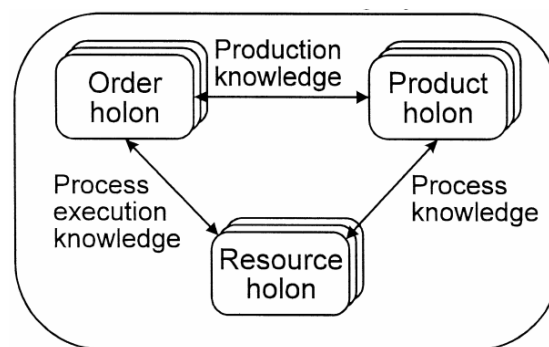
Christensen (1994) derived the following key architectural requirements for holonic system architectures:

- Disturbance handling, availability, robustness
  - Provide intelligent system elements for self- and cooperative planning, scheduling, fault recognition, diagnosis and repair.
- Human integration
  - Provide more intuitive, flexible, responsive, user-customizable human interfaces.
  - Provide “intelligent assistants” to augment human intelligence and prevent human error.
- Flexibility
  - Provide greater human control over system configuration and functionality.
  - Provide self-reconfiguration (“metamorphic”) capabilities.
  - Support continuous/incremental changes in roles and relationships of system elements (“fluidity”).

### 2.2.2 PROSA

Van Brussel et al. (1998) saw great promise in the holonic manufacturing paradigm. They created a reference architecture, around the holonic manufacturing concept, called PROSA. PROSA refers to the composing types of holons. The architecture comprises three basic holon types and one organizational holon type.

As noted by Van Brussel et al. (1998), there are three independent manufacturing concerns in industry: resource aspects (the efficiency of the machine), product aspects (the quality of the product being produced) and logistical aspects (demands and deadlines of the customer). To address these concerns, the resource, product and order holon types were created, as illustrated in Figure 4.



**Figure 4: Basic building blocks of a HMS (Van Brussel, et al., 1998)**

A resource holon consists of a physical part and an information processing part. A physical manufacturing resource is contained inside the resource holon. The physical manufacturing part can be a factory floor, tool holders, personnel or conveyors. The informational part has the methods to allocate resources, and the knowledge and procedures to organize, use and control these production resources and drive production.

A product holon contains all the information concerning the product, including the product's lifecycle, design, bill of materials and quality assurance procedures. This holon is also responsible for the quality of the products and ensures that they are assembled correctly. Therefore a product holon acts as an information server for the other holons.

An order holon represents a task in the manufacturing system. It manages the physical product being produced and all the logistical information related to it. An order holon negotiates with the product and resource holons to get the parts produced.

In addition to the above mentioned holons, the staff holon type was defined. The staff holon provides the basic holons with additional information to aid the basic

holons in making the correct decisions. A staff holon has no authority over the other holons and its presence is not required in a holonic manufacturing system (HMS) for the system to work. It only serves as an advisor to the basic holons.

### 2.2.3 ADACOR

ADaptive holonic COntrol aRchitecture (ADACOR) was created in 2006 as an alternative to PROSA and its main focus is on dynamic control and optimization of the system using higher forms of intelligence (Leitão & Restivo, 2006). The holons in ADACOR not only act in response to their environment, but they are also able to take initiative. The architecture defines four different holon types: product, task, operational and supervisor holons. The product, task and operational holons are very similar to the product, resource and order holons found in the PROSA architecture.

The supervisor holon defined by ADACOR has similar properties to that of the staff holon defined by PROSA, but the supervisor holon has additional features. The supervisor holon is tasked with global optimization and management of the other holons. The supervisor holon takes control of the other holons when the system has entered a stable state. It then optimizes the system for the task at hand and until its optimization algorithms have been satisfied (Leitão & Restivo, 2006). This enables ADACOR to be flexible by using a holarchical/heterarchical control structure when disturbances are detected, but rigid using a hierarchical control structure under normal working conditions to optimize the system.

Compared to PROSA, ADACOR is a more complicated control architecture, but it has the advantages of global optimization and dynamic control structures.

### 2.2.4 Ants

As stated before, one of the shortcomings of the PROSA architectures is its lack of global optimization capabilities. To address this issue Valckeneers & Van Brussel (2005) created a new architecture based on PROSA. The normal process, order and operational holons are created as in PROSA, with additional exploring and intention ant holons that are created by the order holon at regular intervals.

The exploring ant holons explore the system and virtually perform the required processing steps to produce a product. All the exploring ants return their respective routes/solutions to the order holon for evaluation. The order holon then tasks the intention ants with the best solution. The intention ants virtually do the work required to build the product using the solution the order holon assigned to them. The intention ants also reserve the resources needed to produce the virtual part. This allows the order holon to calculate a short term forecast of the production line.

The short term forecasts enable the system to optimize its production using predictive heterarchical control (Valckeneers & Van Brussel, 2005). This control design is unfortunately complicated compared to its predecessor, and requires more computational power and programming effort.

### 2.2.5 Example

To better understand holonic control, the following example was created using the three basic holons defined in PROSA. For this example an electrical motor assembly line will be considered. In the assembly line there are various machines/stations each with their own purpose. The stations are connected by a conveyor system and each station, including the conveyor, is represented by a resource holon. The order holon receives a task to build 10 units of motor type A. The order holon then requests all information regarding motor type A from the relevant product holon. The order holon then checks what assembly steps are required to assemble 10 units of motor type A. Some of the steps would for instance be to wind a rotor, transport parts, assemble a stator, and assemble the rotor and stator. The order holon would then enquire which resource holons are capable of doing the required steps and would then task them respectively.

The order holon only knows what functions the resource holons can perform and which of them are required to assemble the motors. Resource holons only know how to do the required task and not why they are doing it. Only a station's resource holon has knowledge about the physical steps that need to be performed to fulfil the station's function. Because the process information is divided in this manner, the system can easily be reconfigured to build different motors.

## 2.3 Reconfigurable Manufacturing Systems

Rapidly changing technologies and products have forced manufacturers to become more flexible and responsive to keep up with demand (Leitão, 2009).

According to Koren et al. (1999), an RMS is designed from the start to be flexible and modular in order to ensure that the system is future-proof. This allows the manufacturers that used RMSs to change subsystems and parts of a production line enabling it to manufacture a new range of products with minimum down time and cost (Koren et al., 1999).

Koren et al. (1999) also noted that an RMS does not only have to be reconfigurable in hardware, but also in software. The control system has to be able to adapt to new products and subsystems. The control system also has to be able to communicate with other subsystems and diagnose possible problems.

Ideally, an RMS should exhibit the six core characteristics given by (Koren & Shpitalni, 2010):

- Customization: flexibility limited to a part family.
- Convertibility: designed for future functionality changes.
- Scalability: designed for future capacity changes.
- Modularity: system comprises distinct modules.
- Integrability: modules have interfaces suited for rapid integration.
- Diagnosability: designed for easy diagnostics.

The characteristics of an RMS are close to the characteristics of a HMS and therefore holonic control architectures are often used by researchers to create an RMS (Tönshoff & Winkler, 1996) (Heikkilä et al., 1997) (Van Brussel et al., 1998) (Brückner et al., 1998) (Liu et al., 2000) (Chirn & McFarlane, 2000) (Monch et al., 2003).

## 2.4 Control Software Implementation

The following section discusses the different approaches that are commonly used in industry and research environments to implement the control architectures described in the previous sections.

### 2.4.1 IEC 61131-3

In the manufacturing industry all around the world the predominant control approaches are hierarchal and centralized. In these control approaches the controllers are mostly PLCs. PLCs are widely used because they are easy to program and have been proven to be robust in industrial environments. The reason they are easy to program is because the programming languages are standardized by the IEC 61131-3 standard. Maintenance crews can easily learn how to program a PLC and this allows them to fix programming faults in case of system malfunctions. However, code running on machines in industry can often become overcomplicated because the program has to include knowledge of all subsystems and/or devices, even though the coding is simple in principle.

The control architectures mentioned in Section 2.1 are commonly used in manufacturing systems. However, development of these systems requires prior knowledge of the products that they produce/assemble. Once the system is fully developed, it would then be delivered to the client and remain largely unchanged until the product/assembly is discontinued. If the client wished to add or replace a product, the system would have to halt production and people with expert knowledge of the program and system would have to reprogram and modify it. This is often extremely expensive and therefore the company often would rather buy a complete new system to complement or replace the old one.

### 2.4.2 IEC 61499

IEC 61499, also known as Function Blocks, is an event driven programming language. Software that adheres to this standard would be able to create virtual blocks that encapsulate functions, thus the name Function Blocks. These blocks are linked to each other by user defined inputs and outputs. The blocks are triggered by events and when triggered they execute the code inside the block. A function block usually represents a task and can create more events to trigger other blocks if necessary.

Due to the lack of support and debugging tools this standard is rarely used and research on it is limited (Leitão, 2009).

### 2.4.3 Agent Based Control

An agent, as used in the software industry, is a piece of software that is able to make decisions or take actions depending on a specific situation to reach its goals (Vrba, 2012). Agents work together and communicate with each other to complete tasks or solve problems that they would not be able to do without each other's help. Because of the autonomy, modularity, intelligence and cooperation of agents, agent based control is one of the preferred methods for researchers to implement holonic control (Duffie & Piper, 1986) (Maturana & Norrie, 1996) (Van Dyke Parunak, et al., 1998) (Brückner et al., 1998) (Valckenaers et al., 1999) (Monostori & Kadar, 1999) (Liu et al., 2000) (Chirn & McFarlane, 2000) (Sauter & Massotte, 2001) (Monch et al., 2003) (Vrba & Marik, 2005) (Leitão & Restivo, 2006) (Albadawi et al., 2006) (Vrba, 2012).

The Foundation of Intelligent Physical Agents (FIPA) was formed in 1996 to produce standards for agents and multi-agent systems. In 2005 the standards created by FIPA was accepted by the Institute of Electrical and Electronic Engineers (IEEE) and forms part of its standards (The Foundation for Intelligent Physical Agents, 2012). The FIPA standards cover intercommunication of agents, as well as communication of agents with other software.

Almeida et al. (2010) considered the adoption issues of agent based systems and one of these issues was complexity. Agent software is commonly written in languages like Java or Python, which has poor hardware interfaces. Therefore the agent software often requires additional software to interface with the hardware. Additional complications are created by the fact that agent development software written in languages like Java requires a runtime environment to run the agent program. The complex hardware and software make it difficult for maintenance crews in industry to maintain the system.

## 2.5 Evaluation of Control Architecture and Platforms

The following section is an evaluation of control approaches that can possibly be used to create an industry accepted reconfigurable manufacturing control system in the context of station control. Since the holonic control architecture is highly compatible with the characteristics of an RMS as seen in the literature review, and the holonic control is a commonly used approach by researchers for creating an RMS, the use of the holonic control approach for the control of the subsystem of the RMS is assumed here and evaluation of the following architecture and platforms will therefore be done for an ADACOR architecture.

### 2.5.1 IEC61131-3

As mentioned in Section 2.4.1, IEC 61131-3 is the industry standard PLC programming language. It is easy to program and allows direct real time access to hardware inputs and outputs.

In ADACOR, task holons are instantiated dynamically and therefore the control platform needs to be capable of creating dynamic instances of a program. Additionally the control platform also needs to be able to run the individual holons in separate programming threads.

The limitation of IEC61131-3 is its inability to dynamically create instances, as well as the lack of multi-threaded programming support. Additionally the hardware of the PLCs that IEC61131-3 is normally run on, has limited memory and processing power.

One of the characteristics of an RMS is modularity, requiring that software is written in distinct modules. This would mean that each type of holon needs to have its own instance. This in turn requires each type of holon to be written in its own program. Therefore holons cannot be created in IEC 61131-3 without the ability to run multiple threads at the same time.

### 2.5.2 3<sup>rd</sup> Party extensions to IEC61131-3

The OEMs of PLCs have identified the need for more advanced control for PLCs. Even though IEC 61131-3 has object orientated extensions, the OEMs want to allow the industry more freedom. Each OEM's approach differs, but they all seem to be moving in the same direction. Most of the major industrial control OEMs now allows C++ as a programming language for their industrial controllers (Bosch Rexroth, 2013) (Beckhoff, 2013) (Siemens, 2013). The way C++ is implemented differs between the manufactures, but can be divided into three categories.

The first approach uses C++ as a replacement for IEC 61131-3. The second approach has the C++ programs running on a PC separate from the industrial

controller. The industrial controller still uses IEC 61131-3, but the C++ program has direct access to the variables and runtime of the industrial controller. The third approach allows the industrial controller to run the IEC61131-3 programs in parallel with C++ programs on the same controller. This allows the C++ programs to replace or assist the IEC 61131-3 programs.

The result, however, is more complex control software development. Many additional libraries and code have to be used to allow the C++ code access to the controller's inputs and outputs. This in turn requires a deeper understanding of the controller and its inner workings, and therefore training in that specific OEM's implementation of C++.

Therefore, the C++ extensions has to be carefully implemented and used for sections of the control system that will benefit from the advanced programming capabilities, like the supervisor holon. C++ should, however, not be used for sections that might be edited by maintenance staff, who would be unable to change the C++ code.

### 2.5.3 Embedded PC

Embedded PCs were developed to address the shortcomings of PLCs, while still providing the same form factor. Embedded PCs fall between industrial PCs and PLCs. Embedded PCs typically run modified, stripped down versions of the Microsoft Windows operating system along with the OEM's software. In comparison to PLCs, embedded PCs have much more powerful processors and more memory. This allows embedded PCs a much quicker response time and the ability to control more IO than PLCs.

In order to run a PLC program on a Beckhoff controller, a task needs to be assigned to that program. A Beckhoff embedded PC is capable of running multiple tasks. The properties of each task can be changed individually by the programmer. These properties include the number of cycles, the cycle time and the priority of the task. The software runs each task for the number of cycles defined in its properties before running the next task. If one task encounters an error and stops the other tasks are unaffected. The priority defines the order in which the tasks are run. The software allocates and manages the memory for each task. This allows one embedded PC to replace multiple PLCs and extends the capabilities of IEC 61131-3. This effectively means that IEC61131-3 can be used to create holons in separate modules.

### 2.5.4 Agent Platforms

Even though Agent Based Control is the preferred control approach by researchers, its advantages are dulled and its disadvantages emphasised when it is used for station control. ABC is widely used in research to create RMSs, but the applications of ABC in the research community are normally on a cell controller



level. On cell controller level, ABC has little to no hardware interfaces and all the agents or holons exist purely in software. This suits ABC's soft real time environment. In contrast, a station requires hard real time monitoring and execution of IO and parameters. The holons in a station based controller are also directly linked to hardware.

Depending on the level of reconfiguration (Hoffman et al., 2013) an ABC approach can be warranted. However, ABC is still relatively complex and expert knowledge is required to make changes to it. ABC has been extensively researched and has not found industry acceptance. As the main focus of this thesis is to create a control approach that the industry will accept, ABC will not be investigated further.

## 3 Case study

CBI Electric: Low Voltage, the South African industry partner supporting this research, assembles circuit breakers in a factory in Lesotho. Most of the assembly operations are done manually due to the large product variety compared to modest (by international standards) production volumes. However, CBI is considering automating some of its assembly and testing operations with the objective of improving consistency of quality.

The Mechatronics, Automation and Design Research Group of the University of Stellenbosch is developing a Reconfigurable Quality Assurance (RQA) cell for CBI that will improve the consistency of quality of some of their products. The RQA cell will do visual inspections, electrical testing, assembly and labelling of circuit breakers. This cell will be used for various research topics of the research group. The case study for this thesis is the Electronic Test Station (ETS) of the RQA cell.

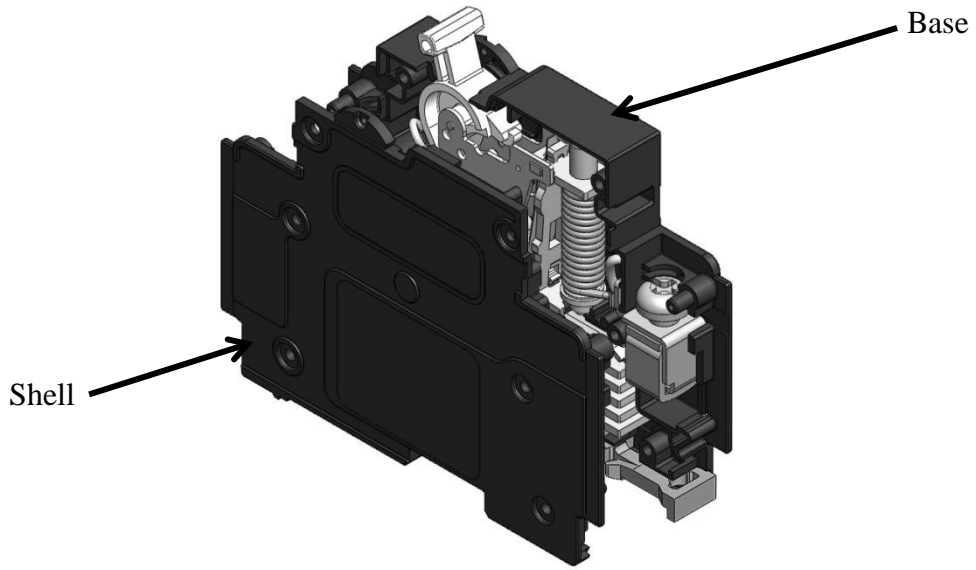
This chapter first describes the cell as a whole and then considers the testing station itself. Thereafter each of the main subsystems of the test station is considered in greater detail.

### 3.1 Reconfigurable Quality Assurance Cell Overview

#### 3.1.1 Product Description

The RQA cell is initially aimed at the Q-frame product family of CBI. The Q-frame has various configurations and versions, which differ in tripping current, tripping curves and small changes in dimensions. Figure 5 shows an exploded view of a Q-frame circuit breaker. Additionally the breakers can also be stacked into multi-pole breakers. Up to four breakers can be stacked on top of each other and are connected internally. These stacked breakers would then work together as one breaker.

In future the RQA cell will be reconfigured to also test other CBI circuit breakers. Figure 6 shows a sample of CBI circuit breakers to illustrate the product variety.



**Figure 5: Exploded view of Q-frame circuit breaker**



**Figure 6: Product variety**

### 3.1.2 Cell Design Requirements

Since CBI places a high priority on quality, every single breaker that they produce is tested. Therefore one of the main requirements of the RQA cell is that it has to be able to keep up with the production of Q-frame circuit breakers, i.e. the cycle time of the cell has to be one breaker per second.

CBI requires the ability to turn off the advanced automated control and to be able control the subsystems using HMIs. This is necessary because they occasionally have to do small ad hoc batches of specialist products. Another factor is that the assembly plant is in a rural area, more than 300 km away from the technical support and, in the event of a fault occurring in the system, they want to be able to continue production.

CBI is developing new electronic testers that allow faster and more accurate testing than conventional methods. These testers are called Ramp Wave Testers (RWT) and CBI wants to use these testers in the ETS. The specific design criteria of the RWT is discussed in Section 3.3.1

To reduce scrap and enable easy reworking of circuit breakers that fail the electrical test, CBI only wants to rivet breakers that have passed the electrical test. This means that the breakers will be tested individually as single pole breakers and that the breakers can fall apart if handled or moved incorrectly. Therefore the ETS has to be designed to always ensure that the breaker stays closed.

CBI's major concern is quality assurance for their circuit breakers. This means that they wanted traceability of every circuit breaker that they produce. They therefore want to retain records of every circuit breaker's test results and a picture of the internals of the every circuit breaker that goes through the cell. To ensure repeatability and traceability of the circuit breakers the whole RQA cell should be fully autonomous with no human operators present.

The Visual Inspection Station requires the shell of the breaker to be off when checking whether all parts are present. In Figure 5 it can be seen that the shell covers all the internal parts. Therefore the shell needs to be placed after the visual station. However the task of placing the shell is too difficult for a robot to do and the placing of the shells will therefore be done by human operators. After the placing of the shell the system is fully autonomous unless a manual override of a station is in effect.

### 3.1.3 Physical Cell Architecture

After an analysis of the functions that must be performed by the RQA cell, the functions were all allocated to a range of stations. The resulting cell layout is shown in Figure 7. The following is a list of the stations in the RQA cell in the order in which they will be used in the cell, as well as a description of each one's functions.

- **Conveyors:** The Conveyors are responsible for transporting the circuit breakers through the system and between the stations.
- **Manual Placing Station:** The circuit breakers are assembled partially, i.e. everything except the shell and clip-in, and the resulting base assemblies are placed on the fixtures on the pallets.
- **Visual Inspection Station:** This station is presented with base assemblies. A machine vision system is used to check for completion of parts, as well as capturing an image for CBI's records.
- **Place Shell Station:** At this station the shells of the circuit breakers are placed on the base assemblies. Because of the difficulty involved in placing the complex part with multiple alignment features, the placing of the shell is done by a human operator. Checks are in place to ensure that the operator places the correct shell on the base assembly.
- **Print Station 1:** This station prints general information related to the circuit breakers, as well as the unique ID of the each circuit breaker in the side of the breaker. The printing is done using a laser printer.
- **Electronic Test Station:** This station is tasked with the electrical testing of the circuit breakers. Every circuit breaker has to be tested by a RWT. This station will be the focus of this thesis and discussed in greater detail later.
- **Stacking and Riveting Station:** This station is responsible for first stacking the circuit breakers correctly and then riveting the stacked set together. If a circuit breaker is not part of a stack it is simply riveted.
- **Printing Station 2:** This printing station is tasked with printing information related to each circuit breaker on the front face of the circuit breaker. This information includes the specifications of the circuit breaker as well as the company's logo.
- **Manual Inspection Station:** The completed circuit breakers are removed from the system and visually inspected by human operators. The clip-ins are also inserted into the completed breakers by the operators.

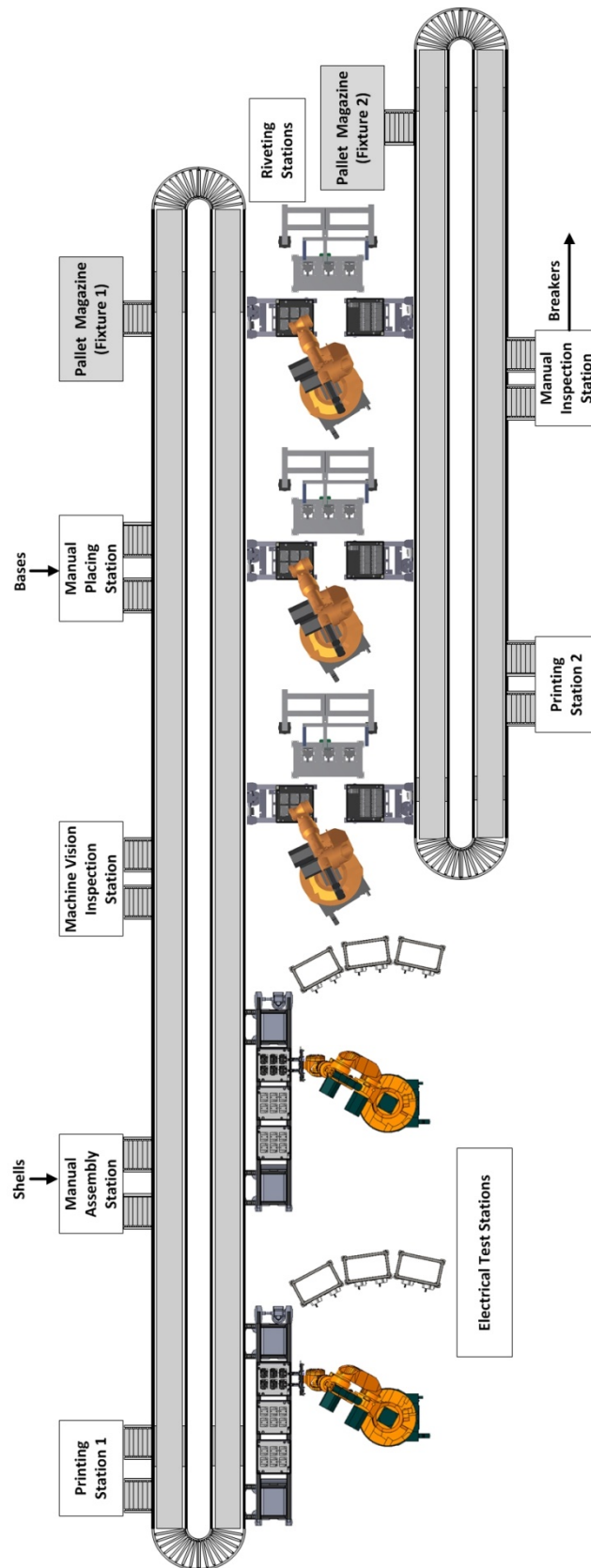


Figure 7: Layout of RQA cell

### 3.1.4 Cell Control Architecture

The RQA cell will be controlled by the cell controller. A holonic architecture based on ADACOR was chosen, because of the global optimization abilities of ADACOR and the importance of cycle time in the cell. Each station will be represented by an operational holon. Each operational holon will manage its own station, but, since CBI requires each station to be capable of a manual override, each station should also allow for a human to replace the interface between the cell controller and the station.

The cell controller will communicate with operational holons using TCP/IP and XML formatted strings. TCP/IP with XML formatted strings was chosen as the communication protocol. The cell controller will send messages to the electronic test station's controller regarding new tasks and pallet positions. The cell controller will coordinate with the conveyor controller to move the pallets. Therefore, the cell controller is also responsible for keeping track of the pallets, as well as what is on each of the pallets.

## 3.2 Electronic Test Station Overview

The ETS is the main testing station of the RQA cell. It is responsible for electrical testing of all the circuit breakers that go through the RQA cell. In this section the ETS, its design criteria and mechanical aspects are described in more detail. The control aspects, which are the main focus of this thesis, are discussed in Chapter 4

### 3.2.1 Station Design Requirements (Performance and Functional)

The following design requirements were derived from the RQA cell's design requirements (Section 3.1.2).

- The ETS has to use CBI's RWTs to test the circuit breakers.
- The ETS has to be able to work with unriveted single circuit breakers. To handle a riveted circuit breaker care has to be taken to ensure that they do not open accidentally.
- The ETS has to allow manual override of the control system. In such a situation the RWTs has to be accessible to humans to continue production. This affects not only the control system, but also the hardware design of the ETS.
- The ETS has to be capable of testing breakers at a cycle time of one breaker per second.

- Breakers that fail the electrical test have to be put in a rework shoot and not with the breakers that passed the test.
- The ETS should be capable of picking the breakers up from a pallet and be able to insert the circuit breakers into the test slots of the RWTs. The breakers have to be transported lying flat due to the stations before and after the ETS, but the breakers have to be tested upright in the RWTs.
- Lastly the ETS has to keep track of the test results of the individual breakers and report the results to the RQA cell controller.

### 3.2.2 Station Hardware Architecture

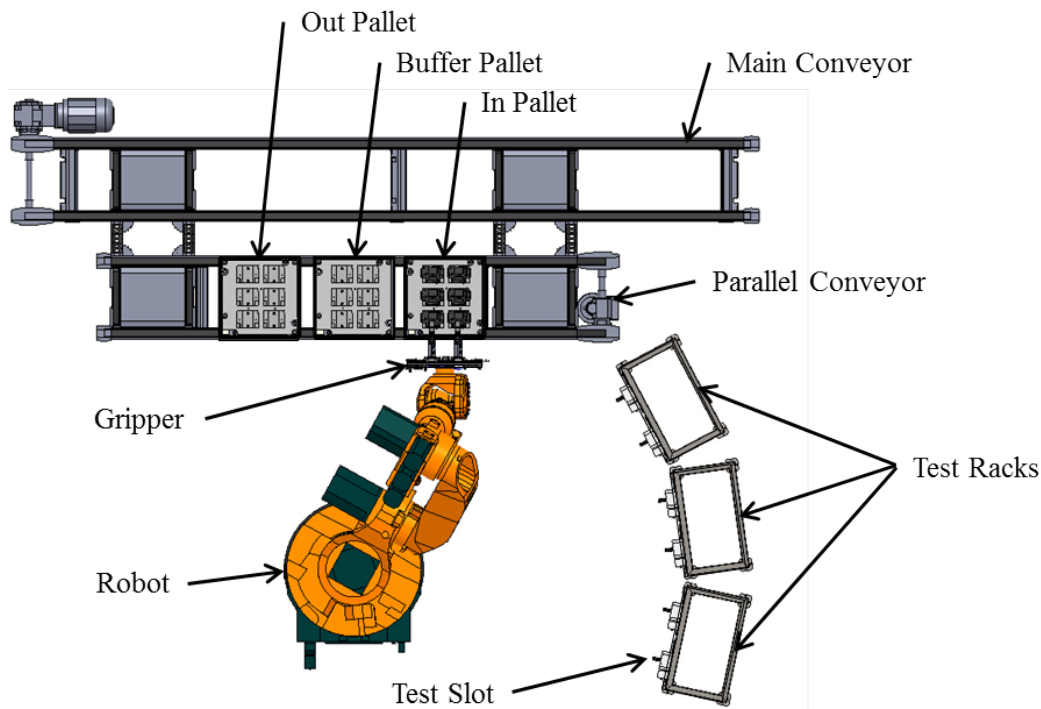
Figure 8 shows the layout of the ETS. A six degree of freedom robot was chosen to transport the circuit breakers between the pallets and the RWTs. The motivation for selecting the robot is discussed in Section 3.3.2. A parallel conveyor would be added to the central conveyor line to allow buffering of pallets while the robot picks and places the circuit breakers of the pallets. More detail about each of the subsystems of the ETS will be discussed below. To achieve the cycle time requirement, an analysis of the throughput of the ETS indicates that two ETSs will be required.

Weighing and scanning were considered as possible subsystems of the ETS. Since Printing Station 1 prints the unique IDs of the circuit breakers on their side, the scanning subsystem would only serve as a check of the printing. The purpose of the weighing subsystem is to check for missing components that cannot be detected by visual inspection. The weight of these components, however, is too little to be detected confidently, since the fluctuations in weight of the larger components can be more than the weight of the small components. In addition to these factors, the scanning and weighing subsystems would have a large impact on the cycle time of the ETS.

The layout of the ETS, as seen in Figure 8, is largely determined by the manual override requirement placed by CBI. This requirement forces the RWTs to be a certain distance above the floor. Further, space must be provided where the operator would stand. Without this requirement, the unused space in the station can be reduced and smaller, faster robots can be used.

The alternatives considered, as well as the design criteria of the implemented subsystems of the ETS, are discussed in their respective sections below.





**Figure 8: Layout of the ETS**

### 3.3 Electronic Test Station Subsystems

This section describes the subsystems of the ETS. The control of these subsystems is described in Chapter 4.

#### 3.3.1 Ramp Wave Tester

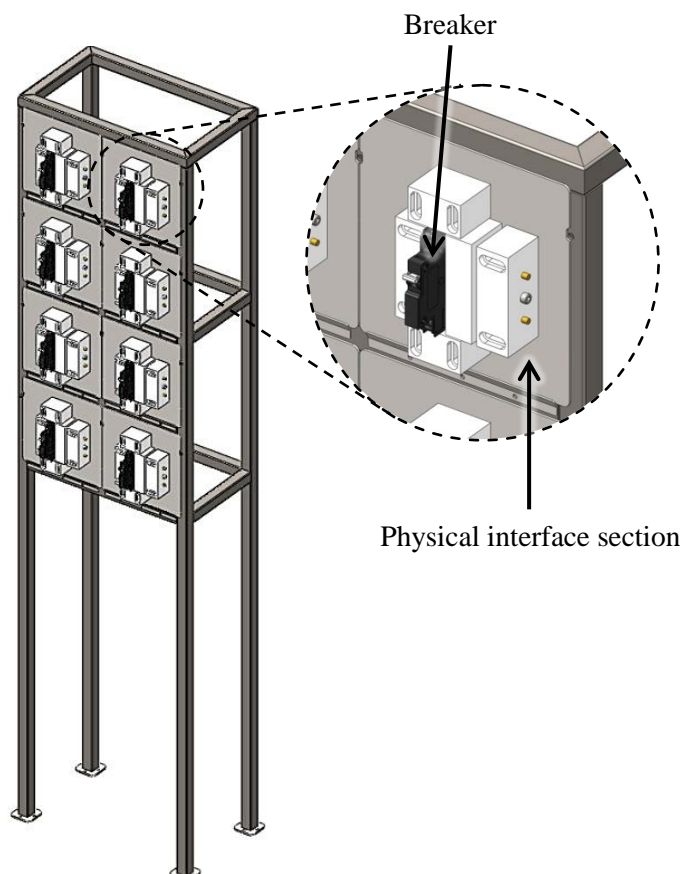
As stated before, the RWTs are designed by CBI. Each Ramp Wave Tester has a physical interface section and an electronics section. The physical interface section contains the test slot, which the breaker is inserted into, actuators and sensors. The electronics section uses the sensors and actuators to test the circuit breaker inserted into the test slot of the physical interface. The electronics section is connected to the physical interface section with a cable and can be up to two meters away from the physical interface section.

This design allows different physical interfaces for different breakers, while keeping the electronics section that does the testing the same. A circuit breaker is not allowed to be within a 100 mm radius of any ferrous metals or other breakers while it is being tested. All the elements of the testing procedure are done by the RWT including switching the breakers on and off. Therefore, only the slots that the breakers fit into and the distance between the breakers had to be taken into account when designing the test racks for the RWTs. The RWTs communicate

with the station controller through RS232, but exact details on the commands are not finalized by CBI. With regards to the communication with the RWTs at the time of writing this, all that is known is that the RWTs await test settings to which they would reply with test results once the tests have been completed.

In order to make the system reconfigurable, test racks were designed with eight positions for physical interface sections of RWTs. The physical interface sections slot into these eight positions and are easily removable and replaceable. This allows operators or technicians to replace faulty RWT interfaces or introduce new physical interfaces that allow new breakers to be tested.

In Figure 9 a test rack mock up with eight physical interfaces can be seen, as well as a close up of one of the physical interface sections. This mock up test rack was built to test the physical interaction of the robot with the test rack. Each physical interface section has a test slot where the breakers are inserted into, to simulate testing. The close up of the physical interface section shows the test slot with a breaker inserted.



**Figure 9: Eight position test rack mock up**

### 3.3.2 Robot

CBI required that the ETS uses the RWTs that they are developing to test the breakers. As stated in Section 3.1.2, CBI also wants the ability to manually override the control system and continue testing without the presence of the control system. This meant that the testers used by the ETS need to be accessible to people when the control system is overridden or faulty.

The robot, in combination with the gripper, should be capable of picking the circuit breakers up from the pallets (Section 3.3.4), where the breakers are on their sides, and placing the breakers into the RWT slots, as well as the reverse operation.

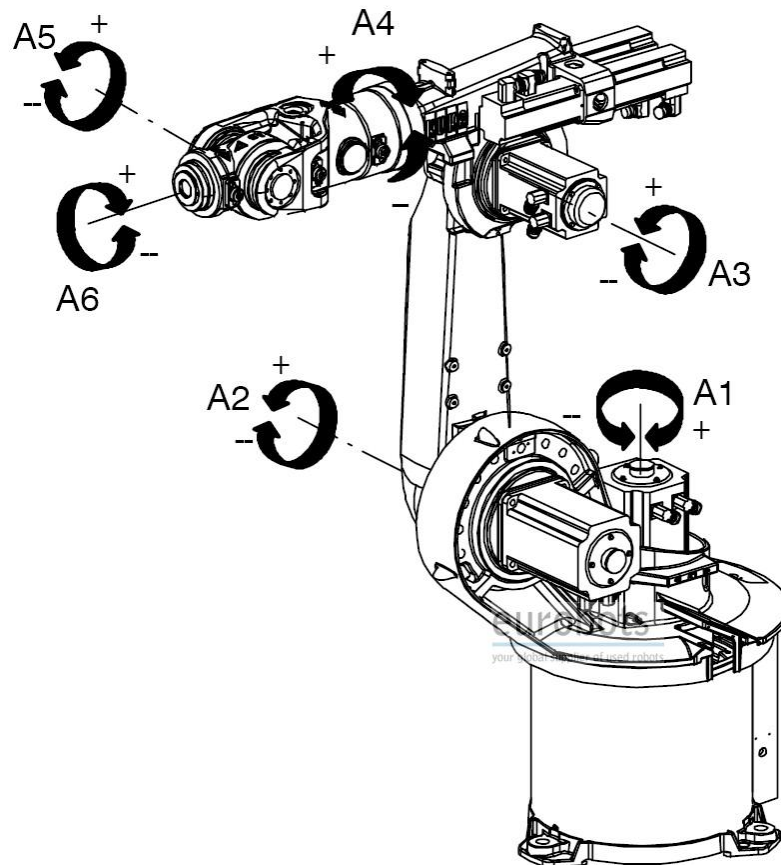
All the traditional types of pick and place robots (SCARA and linear drive based robots) were considered, but only the 6 degree of freedom robot met all the design requirements. Commonly used pick and place robots have four or fewer degrees of freedom, and pick and place from above. These robots cannot be used since the circuit breakers are unriveted and cannot be moved without a mechanism that keeps the circuit breaker closed. These robots also pick and place from above using horizontal work areas. This results in a work area that is inaccessible to humans in case of a manual override. A six degree of freedom robot has a large three dimensional work area and is not constrained to a particular plane. It is also capable of changing the orientation of the end effector in 3 dimensions (roll, pitch and yaw) in comparison to the commonly used pick and place robots that can only change 1 dimension (yaw) of orientation.

The robot chosen to perform the tasks is a KUKA KR16. A smaller, faster robot could be used, but the smaller robot would have a shorter reach and this would lead to a work area that is too small for a human to work in. An alternative approach is to remove the small robot from the work area to provide sufficient space for the human operator to work in, but removing the robot would mean recalibrating the robot when reinstalling it. This can be a costly and time consuming process. The KR16 has a longer reach and its work area can accommodate humans in the event of a manual override.

Another possible alternative would be to use small 6 degree of freedom robots mounted on a seventh axis. The seventh axis would be a linear drive with rails that the robot is mounted on. This would allow the robot to move itself out of the work area, without losing any calibration. This would allow for an optimised work area, but the cost and complexity would increase, particularly since the seventh axis would have to be sturdy enough to meet the accuracy requirements.

Unfortunately, small robots typically have a smaller maximum payload. This limits the gripper design, which in turn affects the number of breakers the robot can move at a time. The gripper design is discussed in Section 3.3.3.

The requirement that the robot's work area must provide for humans access, leads to the main disadvantage of a large 6 degree of freedom robot, i.e. an un-optimized work area. The robot has to use its A1 axis extensively to move circuit breakers to and from the testers. Since the A1 axis is the slowest axis of the robot, the un-optimized work area of the robot has a negative effect on the cycle time of the system. As seen in Figure 10, A1 is the slowest axis because it has to move the largest mass.



**Figure 10: Diagram indicating the different axes of the robot (Eurobots, 2012)**

### 3.3.3 Gripper

One of the biggest concerns of the ETS is cycle time. The target cycle time for the cell is a breaker per second. The gripper, in combination with the robot, therefore has to move the circuit breakers to and from the testers as quickly as possible. The gripper also has to keep the circuit breakers closed when moving them. Additionally, the gripper must be light as not to exceed the payload of the chosen

robot. Lastly, the gripper has to be capable of picking and placing the circuit breakers on and from the pallets and test slots. The orientations of the breakers in the jigs of the pallets are different to the orientation of the breakers in the test slots on the testers.

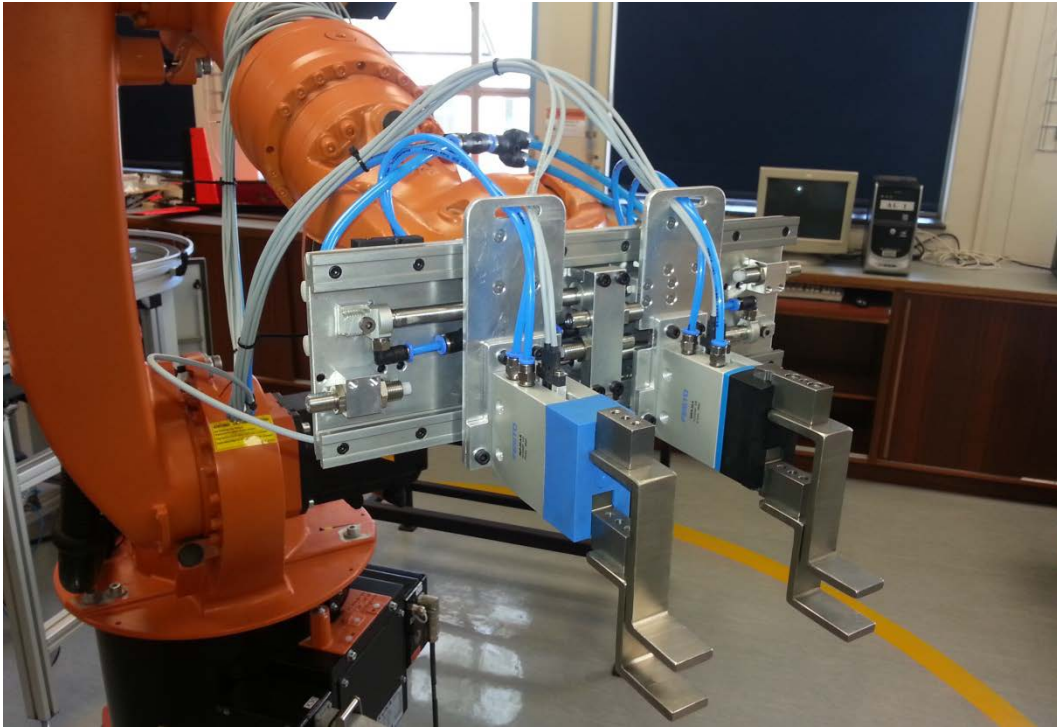
The initial gripper design had a gripper that only picked up one breaker at a time. This allowed great convertibility and robustness. A cycle time estimate however, showed that it would take the robot 8 seconds in total, disregarding the time it takes to test, to move the breaker through the test procedure, which is eight times too slow. The obvious solution is to increase the number of breakers that the robot picks up at a time. However, by using a gripper that picks up multiple breakers at the same time, the system has to place the breakers in neighbouring testers. The robot would also have to wait for all the breakers to finish testing before removing them.

Since the distance between the grippers (pitch) is of great importance when picking and placing multiple breakers at the same time, the gripper would have to compensate for changes in the pitch between the pallet fixtures and the tester slots. In addition to the above mentioned problems, the time the RWTs are idle is also a large concern. This is mainly due to the high cost of the RWTs.

The pallet that the breakers are transported on also affects the design of the gripper. The distance between the breakers, the orientation of the breakers and the layout of the fixtures on the pallet greatly affect the design of the gripper. It therefore comes down to a balance between reconfigurability vs. cycle time and costs.

By using a gripper that picks up a large number of breakers at a time, for example three, the system would need to find three RWT slots that are open, inline and ready for testing. If one of the RWTs becomes faulty, the other two working RWTs will be out of commission until the faulty one is fixed. The advantage of a gripper that can pick up three breakers at a time is a 3 fold reduction in cycle time. Grippers can be designed to overcome the above mentioned flaws, but the cost of the gripper, the cost of the programming to effectively use such a gripper, and the payload limitations of the robot are significant considerations. Such a gripper design would involve using extra actuators that would push the individual gripper jaws forward.

Figure 11 shows the gripper that was designed after considering all the above mentioned aspects. The gripper is capable of picking and placing two breakers at the same time and has an adjustable pitch. The pitch is adjusted using mechanical end stops with built in shock absorbers. The minimum pitch is adjustable between 95 mm and 130 mm and the maximum pitch is adjustable between 190 mm and 250 mm.



**Figure 11: Picture of the designed gripper**

#### 3.3.4 Pallets and Transport System

The transport system used by the RQA cell is a pallet based conveyor system, i.e. a Bosch Rexroth TS2 Plus conveyor. The circuit breakers are transported between stations in groups of six on pallets. The pallets have fixtures mounted on them which secure the circuit breakers in place. When a pallet is in a position where work needs to be done to it, a locator jig is lifted up underneath the pallet locating it accurately. As mentioned in Section 3.1.4, the conveyor is managed by the cell controller.

The conveyor controller will move pallets with breakers that need testing to the ETS. Due to the cycle time requirements, the conveyor will be very busy and congestion will be a concern. The ETS must therefore not cause any congestion of the main conveyor. The station's robot further requires In- and Out-pallet positions to pick and place circuit breakers to and from.

Two transverse conveyors can be used to simply provide the robot with an In-pallet and an Out-pallet. Once the In-pallet is empty, it becomes the Out-pallet and vice versa. The disadvantage to this approach is the possibility of the robot waiting for a pallet. This can happen because this approach has no buffer for the

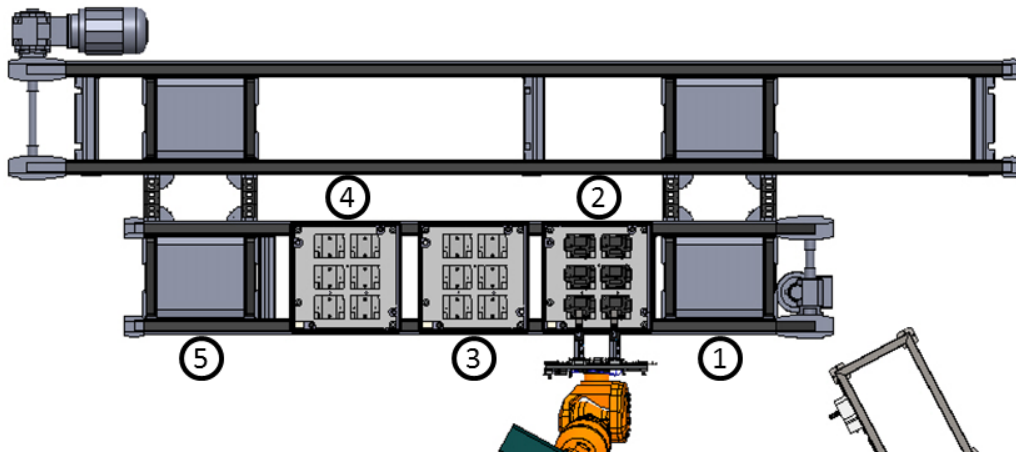
pallets. A buffer can be added at the cost of congestion in the main conveyor, or by adding another transverse conveyor.

Alternatively a parallel conveyor can be used. The parallel conveyor consists of In-, buffer-, and Out-pallet positions. Full pallets that need testing are moved by the main conveyor onto the parallel conveyor. The parallel conveyor shown in Figure 12 ensures that there is always a pallet in position 2 and 4 by making use of 3 buffers.

- In-buffer (1): Buffering a pallet full of untested circuit breakers.
- In-pallet (2): Position where robot picks up untested circuit breakers.
- Buffer (3): Buffering an empty pallet for the Out-pallet position.
- Out-pallet (4): Position where circuit breakers are placed by the robot after successful testing.
- Out-buffer (5): Buffering a pallet full of tested breakers for the main conveyor line.

This configuration ensures that there are always pallets for the robot to pick from and place to. It also reduces congestion of the main conveyor line by buffering pallets full of untested and tested breakers respectively.

The disadvantage of this approach is the costs and complexity involved in adding the extra buffers. It should be noted that buffering only requires space on the conveyor and a pneumatic actuator.



**Figure 12: Overhead view of parallel conveyor and pallet positions**

## **3.4 Reconfigurability Considerations**

The subsystems of the ETS were designed to adhere to the characteristics of an RMS. In this section the physical configuration of the ETS will be evaluated using these characteristics.

### **3.4.1 Customization**

Almost all of the Q-range circuit breakers have the same outer dimensions with only a few small variations. Since changes in the current or voltage of the circuit breakers does not change the outer dimensions of the breakers, new breakers with different tripping current or voltage can be introduced without changing anything other than the test parameters.

### **3.4.2 Convertibility**

The jigs on the pallets and the test slots of the RWTs can be changed when new products are introduced. If the distance between the breakers is changed on the pallet or on the test rack, the gripper can be adjusted to accommodate the changes.

### **3.4.3 Scalability**

New test racks can be introduced to increase the number of testers available to the robot. The gripper on the robot can also be changed to a larger gripper that picks and places more circuit breakers at a time. Further, the entire ETS can be duplicated if a higher throughput is required.

### **3.4.4 Modularity**

The subsystems of the EST have been designed to form distinct modules and are discussed in the relevant sections of this chapter.

### **3.4.5 Integrability**

The physical intergability of the subsystems is high, because a 6 degree of freedom robot is used to move the breakers. This allows the software of the ETS complete control over the movement of the breakers in the station.

### **3.4.6 Diagnosability**

The ETS's ability to detect faulty breakers depends on the RTWs. The ETS checks for the presence of breakers when picking and placing by using sensors in the gripper jaws.



## 4 Control Development

The main motivation of this thesis is to create an RMS that the industry is willing to accept. Even though the ETS is a subsystem of the RQA cell, it was also designed to be reconfigurable internally. The hardware of the ETS has been designed to adhere to both the requirements of an industry partner and the characteristics of an RMS as discussed in Section 3.4. A control system was created to adhere to the same requirements and characteristics. The control hardware and architecture is discussed first, followed by the data that is shared by the holons. The section that describes the data shared between the holons is crucial to the understanding of the rest of the control system. Thereafter the individual holons of the control system is described in detail.

### 4.1 Control Hardware

To maximize the potential of industry adoption, the control system was created using only standard IEC 61131-3 programming language (structured text) running on an embedded PC. Initially, the use of C++ extensions in Beckhoff's platform was considered, but these C++ extensions were found to not contribute sufficiently for the case study to warrant deviating from IEC 61131-3.

As discussed in Section 2.5.3, the embedded PC extends the capabilities of IEC 61131-3, enabling multiple PLC programs to be run in separate threads. This effectively makes a holonic control approach possible. The PLC programs running on the embedded PC have inputs and outputs like normal PLCs. Additionally, the PLCs that run on the embedded PC can have their inputs and outputs linked to each other, thus creating shared memory. These inputs and outputs can be conventional digital IO, but can also be data tables.

### 4.2 Control Architecture

A holonic control approach was used for the control system of the ETS because holonic control has similar characteristics to RMS and holonic control architectures are often used to create RMSs (Section 2.3).

The control system for the ETS is based on the ADACOR control approach. PROSA and ADACOR were considered as possible control architectures, but ADACOR was chosen over PROSA because ADACOR places an emphasis on optimization of the system. The optimization of ADACOR is used in determining the best circuit breakers to pick up and the best location to place them. This potentially has a great effect on the cycle time of the system.

The ETS controller consists of supervisor, product, task and operational holons as described in the ADACOR control architecture. Each type of holon was assigned to a PLC task (Section 2.5.3) to allow the holons to run in separate threads on the embedded PC. The function of each of the holons is described in Section 2.2.3. How the holons were implemented is discussed in the rest of this chapter.

As discussed in Section 3.1.4 the cell controller uses the holonic control approach and will communicate with the ETS's controller using TCP/IP and XML formatted strings. Therefore the ETS's controller would have to be able to send and receive these strings. The cell controller would also only communicate with the ETS using one address and port, because the cell controller views the ETS as an operational holon. This means that all outside communication of the ETS will have to go through a central holon, in this case the supervisor holon (Section 4.5). Figure 13 shows the control architecture of the ETS as well as the communication structure in the ETS. The dashed lines in Figure 13 indicate the controllers used to run the various holons.

A library was created that encapsulates all the basic holon functions. This library makes it possible to easily create new holons and modify all the existing holons. The main difference between the holons was their main program and holon specific functions.

Each test rack in the ETS is managed by a PLC and is viewed as an operational holon (Figure 13). Alternatively the RWTs could each be viewed or programmed as an operational holon into the system.

However, the chosen approach improves the integrability, modularity and diagnosability of the ETS:

- **Integrability:** The ETS's controller only has to communicate with the PLC. The interface will stay the same even if the RWTs change.
- **Modularity:** The test rack, RWTs connected to it, and the PLC forms a distinct module.
- **Diagnosability:** The PLC manages the RWTs connected to it and is capable of fault finding the RWTs if necessary.

The PLC communicates with the RWTs and keeps track of the location of their respective test slots, as well as their status and test parameters. The location of the test slot of each RWT with reference to the test racks origin is stored in a .csv file on the PLC. These locations are used by the supervisor holon to determine which test slots to book. When a test slot is changed, the operator has to update the .csv file on the PLC.

Complete test racks can be reconfigured and tested outside of the ETS and introduced as one unit. This also allows the test racks to be used without the task or supervisor holon when the station is in manual override.

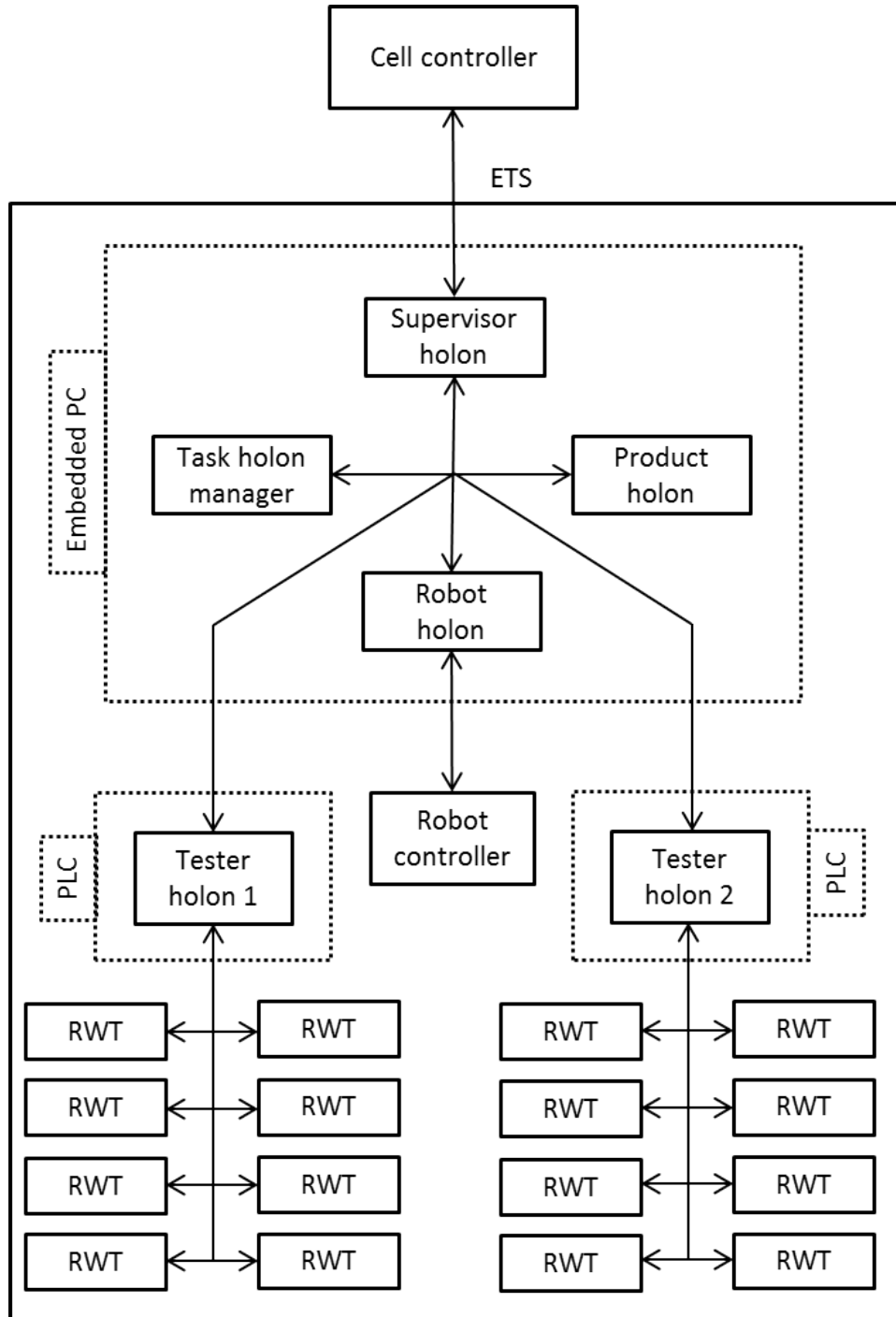


Figure 13: Control architecture of the ETS

When a test rack is moved or a new test rack is introduced, the robot would be calibrated. This calibration would involve matching the origin of the test rack with that of the workspace that the robot designates to the tester. After doing this, all the locations of the test slots can be used by the robot to transport the circuit breakers to and from the testers.

The disadvantages of this approach is the cost of adding an additional PLC, as well as the costs involved in programming the PLC to manage the RWTs and communicate with other holons. However, in the present application, the reconfigurability gained by the additional PLCs offsets the additional cost.

### 4.3 Data Shared Between Holons

#### 4.3.1 Terminology in a Workspace

This section describes the terminology used in the workspaces of the ETS. Each holon that has a physical interface, that the robot can pick and place circuit breakers in, has a workspace. The workspace therefore refers to the space that the operational holon is responsible for. Each workspace has an origin and a local coordinate system that is stored in the robot's controller. Figure 14 shows the difference between a workspace, position ID and location.

Each workspace has several position IDs. A position ID is a unique number assigned to every test slot or fixture in a workspace that the robot can pick or place a breaker in. A position ID number is only unique to the workspace that it is in. Each position ID in a workspace has a location. Location refers to the X-, Y- and Z- coordinates of the position ID measured from the origin of the workspace. The location is always measured in the local coordinate system of the workspace.

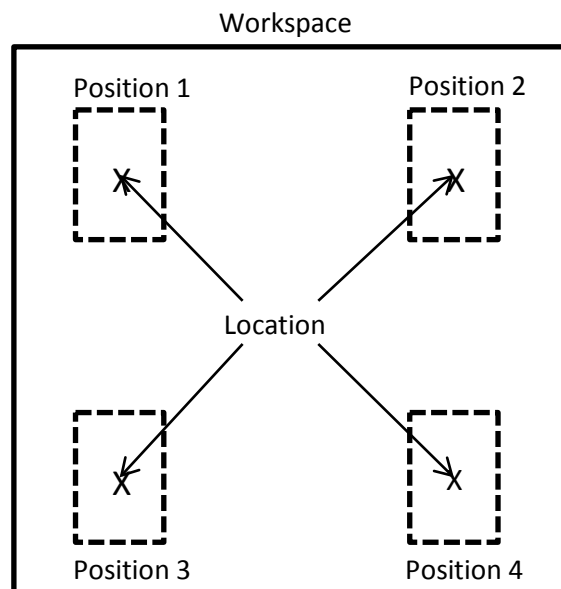


Figure 14: Terminology in a workspace

When reference is made to the position ID of a breaker, it refers to the position ID of the test slot or fixture that the breaker is currently occupying. Similarly, when a reference is made to the location of a breaker, it refers to the location of the test slot or fixture that the breaker is currently occupying. When a reference is made to the location or position ID of a task holon, it refers to the location or position ID of the circuit breaker that the task holon represents. Each holon that has a workspace stores all the information related to its workspace locally in a workspace table.

#### 4.3.2 Workspaces Table

The workspaces table is written and used by the supervisor holon to determine the optimal positions for the breakers to move to. The supervisor holon requests the workspace information from all the holons that have workspaces and stores the information in the workspaces table. The supervisor holon updates the workspaces table every second. The workspaces table is therefore a copy of the workspace information of all the workspaces in the ETS. Table 1 shows an example of the workspace table.

Each workspace has a workspace ID field and an origin field that stores the location of the workspace's origin/datum. The origin field is measured with respect to the robot's world coordinates. Each workspace also has a list of position IDs in the workspace where breakers can be picked or placed, the test settings and test results of each of the position IDs, as well as a booked and resident field which indicate which breaker has booked or is currently occupying a position ID. Position ID and location is discussed in Section 4.3.1.

The origin of each workspace is saved internally in the robot controller when the workspace is calibrated using the robot. The supervisor holon obtains the origin of each workspace from the robot holon. The origin is used by the supervisor holon to locate the best workspace to move breakers to.

When a task holon sends a command to the robot to move a breaker to a specific position ID in a workspace, the task holon would also send a command to the holon that the workspace belongs to and book the position ID it wants to move the breaker to. The supervisor will request an update on the workspace information of the holon. After the holon has replied to the request the workspaces table is updated and will show the position in the workspace has been booked. This allows the supervisor holon to see which position IDs are available. Once a breaker has been moved to its booked position in a workspace, the robot holon will send a command to the holon that the workspace belongs to, to clear the booked field of the position and set the resident field to the breaker's serial number.

**Table 1: Example Workspace Table**

Workspace ID	Origin	Position ID	Test Settings	Test Results	Resident	Booked	Location			
In-pallet	X1217.0 Y187.98 Z595.76 A-89.1 B0.250 C-0.155	1	30A		127		X88	Y23	Z24	
		2	30A		128		X203	Y23	Z24	
		3						X88	Y128	Z24
		4						X203	Y128	Z24
		5						X88	Y233	Z24
		6						X203	Y233	Z24
Out-pallet	X1209.0 Y677.15 Z600.0 A-89.01 B0.34 C-0.360	1					X88	Y23	Z24	
		2					X203	Y23	Z24	
		3					125	X88	Y128	Z24
		4					126	X203	Y128	Z24
		5						X88	Y233	Z24
		6						X203	Y233	Z24
TestRack1	X-212.0 Y1205.0 Z1052.0 A-170.7 B89.66 C-80.65	1	30A	30A	125		X117	Y106	Z39	
		2	30A	30A	126		X117	Y308	Z39	
		3						X333	Y106	Z39
		4	30A				127	X333	Y308	Z39
		5	30A				128	X548	Y106	Z39
		6	30A	30A		123		X548	Y308	Z39
		7	30A			124		X764	Y106	Z39
		8						X764	Y308	Z39

### 4.3.3 Work in Progress Table

Due to the limitation in IEC 61131-3 that does not allow the dynamic creation of instances or allocation of memory, a different method had to be developed to allow dynamic task holons. A table of information related to each task holon was created to simulate dynamic instances. This table is called the Work in Progress Table (WIP table). The size of the table is determined by a variable and can be changed to accommodate more task holons. This variable can only be changed when the controller is in programming mode. The WIP table is saved by the task holon manager and only the task holon manager can make changes to the WIP table. An example of a WIP table is shown in Table 2.

The WIP stores the following data for each task holon: The serial number, product code, location, workspace ID, status, test settings, test results, command string, active command and the unique ID of the last message that the holon sent. The serial number is the unique ID number of the breaker and is used as the ID of the task holon that is responsible for the breaker. The product code identifies the type of the breaker. The location field represents the location of the breaker in the workspace that it is currently located in. The workspace ID is the unique ID that represents a workspace in the system. The status field represents the status of the breaker, i.e. untested, testing or tested. The test settings field represents the electrical test settings by which the breaker is being tested. The test results in part reflect the test results that were returned by the tester holon after the test. The command string is an ASCII string variable that is used to determine the steps or commands that a task holon has to do to successfully test a breaker. Each character represents a command that the task holon must do. The first character in the command string is removed from the command string and moved to the active command field when the task holon has to execute its commands. This first letter is assigned to the active command field until the command is completed. Once the command is completed the process is repeated until the command string is empty. The unique message ID field stores the unique message ID of the last message that the task holon sent. This is used for message traceability (Section 4.4.1).

### 4.3.4 Parity and Priority Table

The parity and priority table (PPT) is used to determine the order in which the task holons are executed, as well as the pairs in which they are to be moved. In addition to this, the PPT also indicates the optimal position, workspace and location to move to for the respective task holons. Table 3 shows an example of the PPT. The serial number field in the PPT indicates the unique ID of the breaker. The parity field indicates which breakers need to be moved together. The pairing of task holons is discussed in Section 4.5.2. The priority field indicates the priority of the breaker. The methods used to determining the priority of the breakers are discussed in Section 4.5.3.

**Table 2: WIP table with example entries**

Serial number	Product code	Location	Workspace ID	Status	Test settings	Test results	Command string	Active command	Unique message ID
123	Q30	X100.0 Y100.0 Z20.0	TestRack1	Tested	30A	30A	CZ		252
124	Q30	X100.0 Y200.0 Z20.0	TestRack1	Testing	30A		CZ	B	253
125	Q30	X300.0 Y100.0 Z20.0	TestRack1	Tested	30A	30A	Z	C	261
126	Q30	X300.0 Y200.0 Z20.0	TestRack1	Tested	30A	30A	Z	C	262
127	Q30	X20.0 Y100.0 Z20.0	In-pallet	Untested	30A		BCZ	A	273
128	Q30	X20.0 Y200.0 Z20.0	In-pallet	Untested	30A		BCZ	A	274



The next optimal position ID, location and workspace ID represent the best position ID, location and workspace that the task holon should move the breaker to in the next move (Section 4.5.4). The PPT is an output of the supervisor holon and is linked as an input to the task holon manager. The task holon manager uses the priority number indicated in the PPT to determine the order in which the task holons are executed.

**Table 3: Parity and priority table with example**

Serial number	Parity	Priority	Optimal position ID	Optimal location	Optimal workspace ID	Status
123	1	0				Tested
124	1	0				Testing
125	2	1	3	X20.0 Y100.0 Z20.0	Out-pallet	Tested
126	2	2	4	X20.0 Y200.0 Z20.0	Out-pallet	Tested
127	3	3	4	X500.0 Y100.0 Z20.0	TestRack1	Untested
128	3	4	5	X500.0 Y200.0 Z20.0	TestRack1	Untested

## 4.4 Interholon Communication

To comply with both the characteristics of holonic control and RMS, the holons need to be distinct modules with interfaces suited for rapid integration. The communication interface also needs to support first in first out (FIFO) buffering, because the holons run in separate threads.

In the controller presented here, the holons communicate with each other by sending XML formatted strings via TCP/IP sockets. Alternatively the holons can communicate with each other using OPC as a central information server. XML formatted strings via TCP/IP was chosen over OPC because the author could easily implement a FIFO buffer and did not have an OPC server available. Further investigation into using OPC as a means to inter-holon communication is being considered.

Each holon has a client and server part. When the holon starts up it reads from a file that contains the IP addresses, ports and device ID's of all the other devices it needs to connect to. This information is then stored internally in the device list table. The holons then attempts to open sockets to all the devices in the device list. The holon attempts to open the sockets every 2 seconds and will create timeout errors if the connection cannot be established in 20 seconds. The holons also starts its multi-client listen server. This means that each holon has a server and a client part, which act like an inbox and outbox, respectively. The server replies to any client that sends messages to it with the same message. The clients each wait for a reply from the server, which they sent messages to, and check that the messages are the same as the messages that they sent. The communication functions have timeouts and error logging buffers to improve the robustness and assist in debugging.

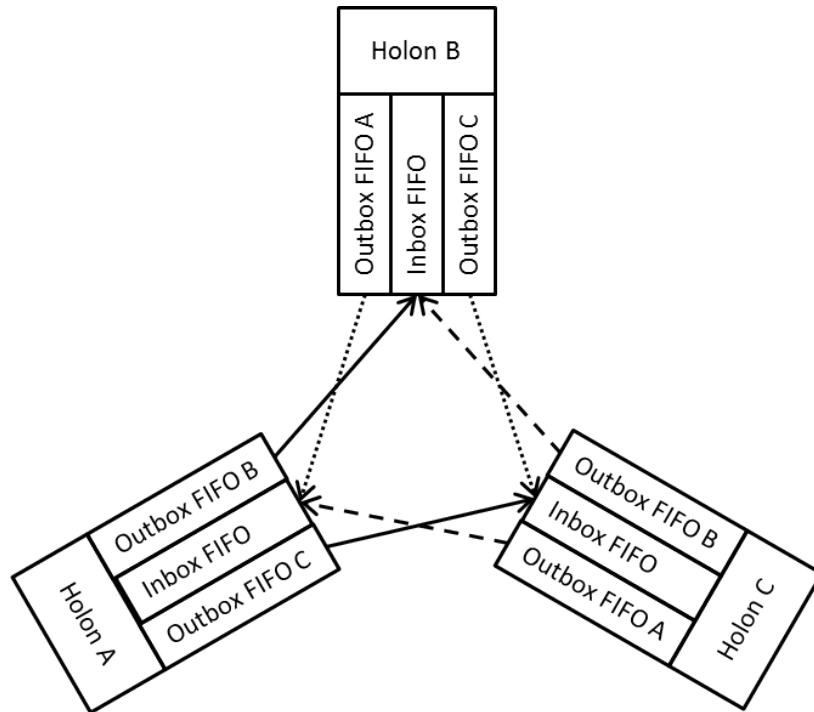
### 4.4.1 Inbox and Outbox Buffers

To make sending and receiving messages to and from different holons possible, the holons make use of FIFO buffers. Each holon has one inbox FIFO that accumulates all the incoming messages from the holon's multi-client server. An outbox FIFO is created for every device in the device list, because each outbox FIFO represents a client connection to a different holon's multi-client server. This means that when holon A wants to send a message to holon B, holon A adds the message to its outbox FIFO B.

The inbox and outbox buffers are checked at the beginning of every program cycle. The messages in the outbox FIFO's will be sent and the inbox FIFO is checked and new incoming messages are interpreted accordingly. Figure 15 shows the link between the outbox FIFO and inbox FIFO of the various holons.

Depending on the type of the received message, certain actions are performed. If the message is a response to a previously sent request, the unique ID of the

message is used to ensure that the correct response is matched with the correct request. A function was written that ensures a unique ID number is generated for each message that the holon sends. This is a crucial feature for the task holons where the individual task holons may send similar requests to the same holon.



**Figure 15: Connection between outbox FIFO and inbox FIFO**

#### 4.4.2 XML Formatting

All the messages sent in the control system has a basic structure that contains a message type, a message, a destination, destination port, the device ID of the sender and a unique ID of the message. The following is an example of an XML formatted message that is sent from a task holon to the product holon:

```
<Message><Type>"ProductInfoReq"</Type><UID>256</UID><Msg>"<PC>1234
546</PC>"</Msg><Dest>"192.168.1.2"</Dest><Port>51200</Port><DevID>"PH"
</DevID></Message>
```

The message type is product information request and the message contains a product code "123456".

XML formatting was chosen because of its wide acceptance in industry and because the cell controller uses it. Care has to be taken when using this formatting with IEC 61131-3 as the maximum string length is 255 characters. The tags of the structure have to therefore be kept small and the sending of unnecessary

information should be avoided. However, if the message exceeds 255 characters, the message can be sent in multiple parts.

## 4.5 Supervisor Holon Implementation

The supervisor holon is responsible for optimising the performance of the ETS. The supervisor holon uses information from the WIP table and Workspaces table to determine which breakers to move, the order in which they need to be moved, as well as where they need to be moved to.

The supervisor holon also serves as a gateway between the internal holons of the ETS and the cell controller (Figure 13). If a holon in the ETS needs to send a message to the cell controller, the message is sent to the supervisor holon. The supervisor holon will then send the message to the cell controller. This is done because the cell controller views the ETS as one operational holon.

The WIP table of the task holon manager is linked to the supervisor holon as an input using the IO linking capabilities described in Section 4.1. This allows the WIP table to act like a direct input to the supervisor holon and greatly reduces the latency of the supervisor holon as it is able to constantly monitor the WIP table. This was done because the supervisor holon decides the order or priority (Section 4.3.4), in which task holons are executed and how they are paired. The supervisor holon pairs new task holons (Section 4.5.2) when they are created, by using the locations of the breakers on the pallet.

The following sections describe how the supervisor holon determines which breakers to move, the order in which they are moved, as well as where they are moved to.

### 4.5.1 Overview

Figure 16 shows the flow chart of the supervisor holon. The supervisor holon first sends and receives all the messages, including the messages to and from the cell controller. The supervisor holon then updates the Workspaces table. When the task holon manager has indicated to the supervisor holon that it needs a new updated PPT, the supervisor holon starts optimising the workspaces of the ETS. The supervisor holon first pairs new task holons that have not been paired yet. The supervisor holon then cycles through the workspaces looking for the breakers that are in the best position ID to be moved. When the supervisor holon has found a breaker, the supervisor holon finds the best position ID to move the breaker to. The supervisor holon then assigns a priority to the breaker and the PPT is updated. Once the supervisor holon has cycled through all the workspaces, it indicates to the task holon manager to use the PPT and activate the designated task holons to move the breakers.

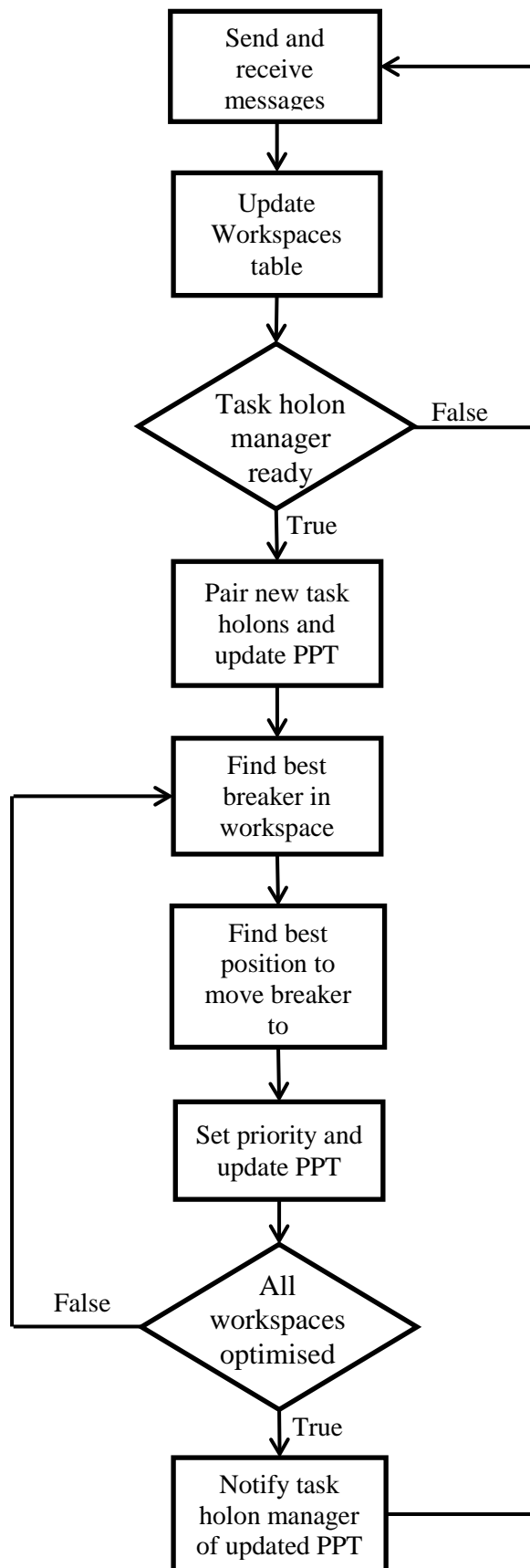


Figure 16: Flow chart of supervisor holon

#### 4.5.2 Pairing of Task Holons

The pairing of the task holons was done to allow the system to move multiple breakers at the same time. The pairing is indicated in the PPT and indicates to the supervisor holon and task holon manager which tasks holons are paired and should be moved together. It should be noted that more than two breakers can be paired together and that the number of breakers with the same parity number is limited to the number of breakers the gripper can pick up at a time. The parity of the task holons is assigned by the supervisor holon when the task holons enter the ETS on the In-pallet. The parity is a unique number that is assigned to task holons that are inline and next to each other. A function was written that ensures a unique parity number is generated for new pairs. Depending on the gripper's design and the layout of the pallets, the supervisor holon will pair the task holons accordingly. The parity would remain unchanged until the task holons leave the ETS. This was done because the robot takes approximately 2 seconds to move from the tester to the Out-pallet and the breakers take a maximum of 30 seconds to test. Reassigning parity dynamically depending on the status of the breakers can cause situations where the gripper is only picking or placing one breaker at a time. This will have a greater detrimental effect on cycle time than simply waiting for a pair to finish testing. If one breaker was to fail the tests done by a RWT, the robot holon would still wait for both to finish testing and pick up both together (Section 4.7).

#### 4.5.3 Priority Assignment

The task holons are executed in the order given by the priority numbers. A priority number of 1 is the highest priority. The supervisor holon assigns priority to the task holons depending on the location (Section 4.3.1) of the breakers that the task holons represent. For the In-pallet and tester workspaces, the supervisor holon assigns the lowest available priority number to the task holon that is in the position ID closest to the origin of the workspace that the task holon is currently located in. The supervisor holon would assign consecutive priority numbers to the task holon in the best position ID and its pairs.

#### 4.5.4 Determining the Best Position to Move to

The best or optimal position ID that a breaker needs to move to is determined by analysing the available workspaces and position IDs. If a breaker has to be tested, the supervisor holon first finds a test rack with open position IDs that is closest to the pallets. The supervisor holon then finds the best position ID on the chosen test rack to move the breaker to. The best position ID is determined by the location of the position IDs. The supervisor holon ensures that the position ID is not occupied, booked or suggested as an optimal position ID for any other task holon. The suggested position IDs is also checked to ensure that the gripper can place the pair simultaneously.

#### 4.5.5 Pallet Holon Functionality

Initially, it was considered to associate an individual holon with the each of the pallet positions (Figure 12). However, their functionality was rather built into the supervisor holon because of the following reasons: The control of all the conveyors in the RQA cell is done by the conveyor controller, which is separate from the ETS. Since all communication to holons outside the ETS goes through the supervisor holon, the messages of the pallet holons would have to be sent by the supervisor holon in any case. With no physical interface or intelligence, the pallet holons would simply become communication relays with data tables. This has no advantage in terms of reconfigurability. Therefore they were integrated into the supervisor holon. Additionally, the supervisor holon requires regular updates on the workspaces and by integrating the pallet holons into the supervisor holon, the communication overhead is reduced and the information is updated quicker.

When a pallet enters the ETS with circuit breakers that need to be tested, the cell controller sends messages that indicate the locations and position IDs on the pallet, as well as the position IDs of the breakers on the relevant pallet. The supervisor holon saves the location information along with the pallets unique ID in the Workspaces table. The information related to a pallet is saved until the pallet leaves the ETS.

### 4.6 Task Holon Manager

The task holons drive production. In the ADACOR architecture, the supervisor holon optimises the RMS by determining which task holon needs to be prioritised over the other task holons. Only the task holon with the highest priority is allowed to send messages and negotiate with the other holons. This means that there is only one task holon sending messages and negotiating with the other holons at any given time. This task holon is called the active task holon. The information of the non-active task holons is updated every cycle if updated information is available.

To optimise cycle time, the gripper on the robot can transport two breakers at the same time (Section 3.3.3). This means that two breakers and therefore two task holons need to be paired together. The supervisor holon determines the pairing of the breakers and is discussed in Section 4.5.2. A task holon manager would therefore be necessary to ensure that the task holons move together and are only moved if both holons have the same status.

As mentioned in Section 4.3.3, all information regarding a task holon is saved in the WIP table. The variables related to the active task holon is loaded from the WIP table by the task holon manager, which is then able to drive production. Since the supervisor holon decides which task holon is active, and since only one

task holon can be active at any given time, the need for dynamic instances for task holons has been avoided.

Since dynamic instances are not possible, an alternative to the approach described above would be to create the same number of task holon instances as there are entries into the WIP table. These instances would each have their own messaging service, IP address and port. These task holons would also have to use variables similar to the entries in the WIP table in order to simulate dynamic instances of task holons. This approach would be more complicated to implement and requires more memory. This approach would also result in more communication overhead.

The task holon manager with a WIP table can successfully simulate task holons because the task holons are relatively simple and only one task holon can be active at a time. Although the task holons are not separate modules, the other holons in the ETS are not aware of this and communicate with the task holons in the same way as they would if the task holons were separate modules.

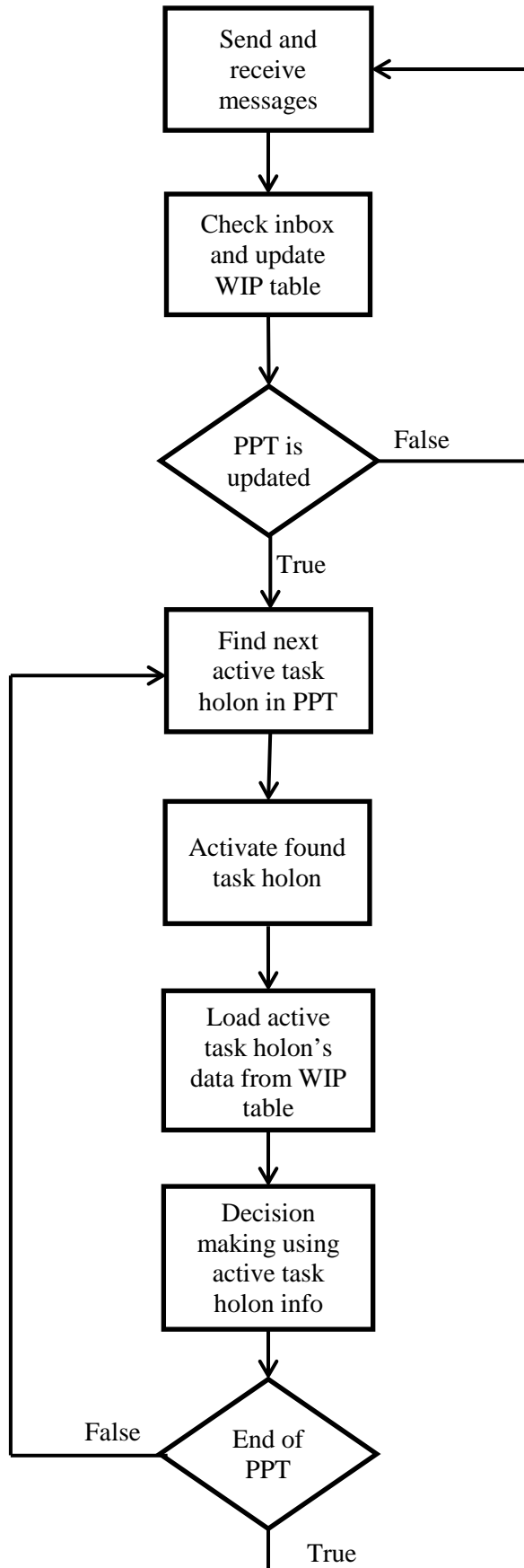
The task holon manager has several parts like any other holon. It consists of a communications part that all the task holons use to communicate (holon communication section), an information processing part where the information related to the task holons is stored and updated, and a decision making part where the task holons execute their commands. Figure 17 shows a flow chart of the task holon manager.

The task holon manager updates the WIP table by checking the Inbox. If the message type is a reply to a request sent by one of the task holons, the relevant task holon is updated. This is done by using the unique message field of the WIP table that keeps track of the messages that the task holons have sent.

The task holon manager receives messages from the cell controller (via the supervisor holon) when new task holons should be created along with the information related to the new task holons. When all the required information is present, the task holon manager adds this information to the WIP table thereby creating a new task holon. The task holon manager will only create a new task holon if it has received the serial number, product code, position ID and workspace ID of the new task holon from the cell controller. The task holon manager then sends a message to the product holon requesting the information needed to successfully test the breaker that the new task holon represents.

The task holon manager then waits until the supervisor holon indicates that the PPT has been updated. When the PPT has been updated, the task holon manager activates the task holons in the order indicated by the PPT. The priority field of the PPT is used to determine the order in which the task holons are activated (Section 4.3.4).





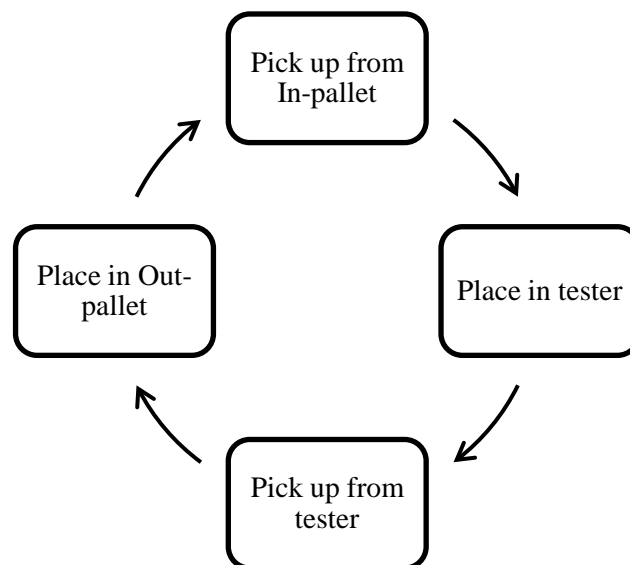
**Figure 17: Flow chart of task holon manager**

The task holons are activated by the task holon manager loading the data relevant to the active task holon from the WIP table and performing actions dictated by the logic stated in the decision making part. The messages that the active task holon wants to send are added to the relevant outbox and will be sent on the next program cycle. The WIP table is updated with the unique message ID of the message that is going to be sent. The next task holon is then activated. The task holon manager will go through the PPT once, activating all the task holons indicated by the PPT. The task holon manager will then wait until the supervisor holon indicates that the PPT has been updated before activating the task holons again.

## 4.7 Robot and Gripper Holon

### 4.7.1 Robot Functionality

To achieve maximum productivity and reduce cycle times, the robot should minimize movements, particularly without breakers in the gripper. The distance the robot has to travel between the In-pallet and Out-pallet is small, similar to the distances that the robot has to move between the test racks, while the distance between the pallets and the test racks is large in comparison. Therefore, in order to improve efficiency and reduce cycle times, the latter distance should always be covered with breakers in the robot's gripper. The optimal movement cycle was therefore found to be: move to In-pallet, pick up from In-pallet, move to tester, place in tester, move to tester (different position), pick up from tester, move to Out-pallet, place in Out-pallet, as illustrated in Figure 18.



**Figure 18: Flow diagram of the robot's movements**

When the robot holon receives new commands from the active task holon, the commands are entered into a table called MoveInfo. The MoveInfo table stores all the information needed to successfully move a breaker and send updated information to the relevant holons. The MoveInfo table has the following data fields: serial number, current location, current workspace ID, destination location, destination workspace ID, parity, In-Buffer flag, scrap flag and unique message ID. The serial number field is used to store the serial number of the breaker. The current location and current workspace ID fields, as well as the desired location and desired workspace ID are self-explanatory. The parity field stores the parity number of the breaker and is used to ensure that only paired breakers are moved. The In-Buffer flag and scrap flag are described in the paragraph below.

When a command is entered into the MoveInfo table, it has not been assigned to one of the buffers that the robot uses to move the breakers. If the move command in the MoveInfo table indicates to move a breaker from the In-pallet, the move command is entered into the InPalletFIFO. If the breaker needs to move to the Out-pallet, the command is entered into the OutPalletFIFO. Because each task holon sends its own command to the robot holon, the robot holon will wait for all the commands of breakers that are paired to be in the MoveInfo table before assigning the commands to one of the buffers. When a command in the MoveInfo table has been assigned to one of the buffers, the robot holon sets the In-Buffer flag of the command entry in the MoveInfo table. The scrap flag is set when a breaker needs to be scrapped. The unique message ID is used when sending messages back to the task holons indicating that the move command has been completed.

As indicated above, the robot holon uses two buffers called InPalletFIFO and OutPalletFIFO. This was done because breakers are either moved from the In-pallet to a tester or from a tester to the Out-pallet, with the exception of a case where a breaker has to be scrapped. If the desired location of the breaker is in the Out-pallet the command is placed in the OutPalletFIFO. Correspondingly, if the current location of the breaker is in the In-pallet the command is put into the InPalletFIFO.

When a breaker has failed, it is also entered into the OutPalletFIFO and the scrap flag is set high. The breaker will then be picked up as normal, but the robot will dump the scrap breaker in the scrap shoot on the way to the Out-pallet. The scrap shoot is placed close to the Out-pallet to keep the cycle time to a minimum. If the scrapped breaker was part of a pair the robot will place the other good/passed breaker in the Out-pallet. If both breakers failed, both will be scrapped and the robot would not move to the Out-pallet. After scrapping of a breaker has occurred, the pair will be broken and an open slot will exist on the Out-pallet. The ETS control system will not try to fill this slot, since the robot would have to move with only one breaker in the gripper to fill the slot and this would be detrimental to the cycle time.

#### 4.7.2 Gripper Functionality

The gripper holon was integrated into the robot holon because any changes to the gripper would result in changes that have to be made the robot holon. All positioning actions or tasks of the robot are programmed by using the tool center point as a frame of reference. This is done to keep the robot from crashing into its workspaces and general safety concerns related to robots. By hardcoding the tool center point into the robot and robot holon, the robot would avoid collisions when invalid move commands have been issued or a software limit has been reached. These safety features are integrated into the robot's controller, but will only be able to work if the tool center point is correctly defined.

This approach leads to the robot holon being in charge of the gripper holons. That also means that the decision making part related to moving breakers is the robot holon's responsibility. Therefore the gripper holon becomes a very basic entity that only has data entries that indicate the locations of the gripper's jaws, how many breakers the gripper can pick up at once, as well a few inputs and outputs. Therefore the gripper holon can be easily integrated into the robot holon. By integrating the gripper holon into the robot holon, changes made to the gripper would automatically be made the robot holon.

An alternative to the abovementioned approach is that the robot holon has an internal holarchy, which can include several grippers. The robot holon then manages these gripper holons separately from the system. If a gripper changer was introduced to the ETS to allow the robot to change grippers dynamically, separate gripper holons would show several advantages over the approach discussed above. To allow grippers to be dynamically added or changed, the tool center point would be set to the point where the grippers are connected to the robot. The robot would then rely on the gripper holon to keep the gripper from crashing into objects.

By integrating the gripper's functionality into the robot, any possible delays caused by interholon communication between the robot and gripper are eliminated. Even though this delay may seem small when compared to the time it takes the robot to move, these small delays can quickly add up. The gripper is actuated six times to move two breakers through the ETS.

Hoffman et al. (2013) discusses the small human effort of a skilled operator to change parameters vs. highly sophisticated control software to enable dynamic changing of the grippers. The intelligence of the control system therefore depends on the level and frequency of reconfigurations.

The only other holon that communicates with the gripper is the supervisor holon. The supervisor holon only needs to know how many breakers the gripper can pick up at a time and where the positions of the jaws of the grippers are. This information is used by the supervisor holon to pair the breakers.

## 4.8 Product Holon

The ETS has its own product holon that stores all product information needed to test the breakers. The ETS's product holon is in constant communication with the cell controller and stores a local copy of all information relevant to the ETS. The information is stored in a .csv file on external or internal storage of the embedded PC. This file can easily be edited using Microsoft Excel. This allows the ETS to continue work even when the cell controller fails or a manual override of the system is activated.

An alternative approach that was considered, but not selected, was to remove the product holon from the ETS. Task holons would then communicate with the cell controller's product holon. This would ensure that the task holons receive the most recently updated information at the cost of higher network traffic to and from the cell controller's product holon. Additionally this would reduce the robustness of the ETS because the ETS will be unable to continue work when the cell controller fails or a manual override of the system is activated.

## 4.9 Tester Holon

The tester holon manages the test rack with the physical interfaces and all the RWTs connected to the test rack. Based on the design described in Section 3.3, each rack has eight RWTs connected to it. The tester holon keeps track of all the relevant information of each RWT using a data table. The data table has the same structure as the workspaces table that was described in Section 4.3.2. The workspace ID and origin will be that of the test rack. The tester holon sends and receives the test settings and test results to and from the RWTs using RS232.

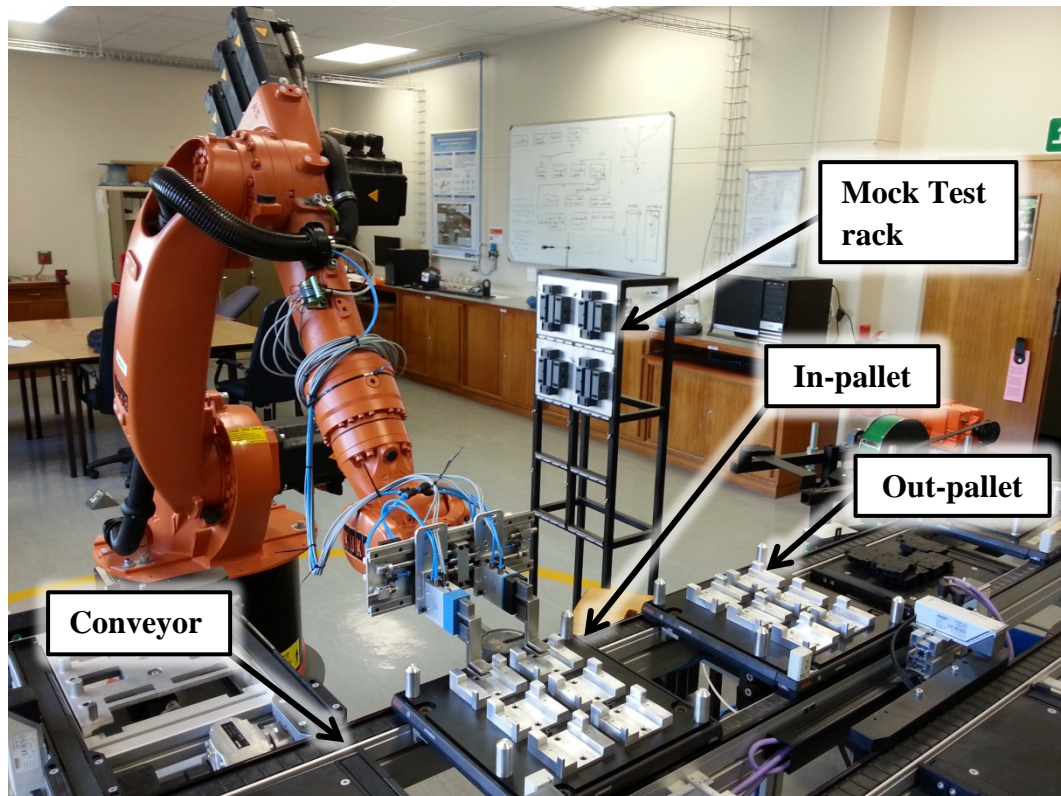
The tester holon therefore enables holonic communication between the task holons and RWTs. This simplifies the control of the RWTs and resolves uncertainties caused by the un-finalized interface of the Ramp Wave Testers.

## 5 Assessment

To achieve the objective stated in Section 1.2, the ETS was assessed using the requirements stated in Section 3.2.1 and the characteristics of an RMS given in Section 2.3. Table 4 shows what RMS characteristics were tested in each assessment. Since the RWTs that test the breakers are still under development, the RWTs functionality was simulated by the tester holons for all the experiments. The physical interfaces of the RWTs were also simulated by building a mock-up test rack with four test slots. The four test slots were used to test the robot's ability to pick and place breakers to and from the testers. A gripper with 2 positions was used for the testing. The cell controller of the RQA cell was under development at the time of testing and the cell controller had to be simulated. To simulate the cell controller, an external PC was used that sends XML formatted strings via TCP/IP. This PC sent commands to the ETS to test breakers in the same way the cell controller would. The external PC did however not have a product holon and therefore the ETS used its own product holon for all product information. Figure 19 shows the laboratory setup of the ETS that was used in this assessment.

**Table 4: RMS characteristics tested by each assessment**

	Customization	Convertibility	Scalability	Modularity	Integrability	Diagnosability
Change within product family	X					
Introduction of new product		X				
Change in origin		X	X			
Change in physical configuration		X	X	X	X	
Change in pallet	X	X		X		
Change in gripper		X	X	X		
Manual override						
Robustness						X
Cycle time			X			
Control evaluation						



**Figure 19: Lab setup of ETS**

## **5.1 Change within Product Family**

To test the customizability of the ETS, an experiment was conducted by changing the test parameters of a type of breaker.

The only change required was to the information saved in the product holon of the ETS and can be done with a medium level of expertise. The experiment was completed successfully in a few minutes and the circuit breakers with the new parameters were tested correctly with the new parameters. This experiment therefore proves that the ETS can accommodate changes within product family as required by the customization criterion of an RMS.

## **5.2 Introduction of a New Product**

To test the convertibility of the ETS, and in particular the control system's ability to accommodate new products, the following experiment was conducted: The test settings, product code and command string of the new product were added to the ETS's product holon for the experiment. Under normal circumstances the

information of the new product would be added to the cell controller's product holon. This information would then be synchronised with the ETS's product holon. The changes to the product holon was done in a few minutes and required a medium level of expertise. Although the reconfiguration was done in a few minutes, the new settings need to be tested and a ramp-up phase is required. Since there were no changes to the physical configuration of the ETS, the ramp-up phase only requires testing of control samples of the new breaker to ensure quality control. This ramp-up phase should be completed in 10 minutes.

For this experiment the dimensions of the product was the same as that of the Q-frame breakers that were tested before. Therefore the same test racks could be used and no changes had to be made to the tester holons. There were also no changes to the gripper. If a new product is introduced with different dimensions which are not compatible with the test racks that are currently installed in the ETS, it would require a change in the physical configuration of the ETS and may also require a change in the gripper. These changes are discussed in Section 5.4 and 5.6 respectively.

The new breakers were tested successfully by testing a total of 100 breakers. This experiment proves the ability of the ETS to successfully introduce new products into the system. If the introduction of a new product requires a change in the physical configuration, and/or a change in gripper, and/or a change in pallet, and/or a change in origin, the individual experiments can be combined to determine the level of expertise, reconfiguration time and ramp-up time required to introduce the new product.

### **5.3 Change in Origin**

The following experiment was done to test the scalability of the ETS. The scalability was tested by changing the origin of a test rack workspace and the In-pallet workspace. The origin of the workspaces was changed and 50 breakers were tested to check if the reconfiguration was successful.

The origin of the test racks and pallets are stored in the robot's controller. To change the origin of a workspace, or create a new workspace, an operator has to use a built-in calibration procedure of the robot. This procedure requires an operator to move the tool centre point of the gripper to 3 points on the new workspace. The 3 points are: the origin, a point on the X axis of the workspace and a point on the XY plane of the workspace. From these 3 points the robot controller calculates the origin and local coordinate system of the workspace. All of the calibration is done by using the pendant (HMI) of the robot's controller.

The origins were changed successfully in 30 minutes and required a medium skill level. After the origins were changed, a ramp-up phase was required to ensure the robot could pick and place breakers successfully in the changed workspaces.



During the ramp-up phase the ETS tests breakers as it normally would with the exception that the robot moves at a greatly reduced speed with a skilled operator monitoring the robot's movement. The pendant of the robot is used by the operator to control the speed of the robot and stop the robot if a possible crash is detected by the operator. This ramp-up phase takes approximately 10 minutes and should be repeated whenever an origin is changed or a new workspace is created.

The experiment was completed successfully with the robot using the new origins. This experiment proves the ability of the ETS to accommodate both changes in the origin of workspaces, as well as the introduction of new workspaces.

#### **5.4 Change in Physical Configuration**

If a new product is introduced with different dimensions which are not compatible with the test racks that are currently installed in the ETS, the physical interfaces of the current racks would have to be changed, or a new test rack would have to be introduced.

If the physical interfaces are changed, the workspace information has to be updated accordingly. This involves a medium level of expertise as the location of each test slot has to be measured accurately and saved on the tester holon. The new measurements can be entered into the tester holon by making use of the test rack's HMI. This reconfiguration can be completed in approximately 30 minutes, but the time varies depending on the number of physical interfaces that have to be changed.

An experiment was conducted to test the ability (convertibility) of the ETS's control system to accommodate a change in the physical interfaces. The locations of two test slots on the mock-up test rack were altered and the new location information was updated in the tester holon associated with the test rack. The test slots were both moved 10 mm on the negative Y axis of the test rack's local coordinate system. The test slots were not moved independently as this would result in a change in gripper which is covered in Section 5.6. A command to test a batch of Q-frame breakers was sent from the cell controller simulator to the ETS. A ramp-up phase similar to the one described in Section 5.3 was done after the changes were made to the test slots to ensure that the robot can pick and place breakers in the test slots without crashing.

The experiment was conducted successfully with the ETS testing 50 breakers. The control system of the ETS adapted to the changes and used the changed test slots successfully.

An additional experiment was done to test the scalability and integrability of the ETS. This was done by adding a new operational holon in the form of a new test rack. To accommodate the new test rack, the device list of all the holons has to be

updated with the address and port of the new test rack. Additionally the robot's controller also has to be updated with the new origin of the new test rack. Since all the holons on the embedded PC use the same device list, the new test rack was added to the device list in a few minutes and required a medium skill level. A new workspace and origin was added for the new test rack as described in Section 5.3. The time and skill required to physically install the test rack was not taken into account as it depends on the design and weight of the test rack. Since only one mock-up test rack was built, the second test rack was added virtually and the ETS was tested in a dry run mode. The dry run means that virtual breakers were used and the sensors in the gripper were modified to always detect a breaker when the grippers are in the closed position.

The experiment was completed successfully with the ETS using two test racks and operating in a dry run mode. A total of 100 breakers were tested successfully during this experiment.

In addition to the experiments, an assessment was done to assess the ability of the ETS in adding a completely new operational holon in the form of a weighing station. The addition of a weighing station would test the integrability, modularity and convertibility of the control system. The weighing station would require changes in several holons. The robot, supervisor, task and product holons would need to be changed to accommodate and use the functionality of a weighing station. In addition to the changes in the holons, the robot's controller would also have to be updated to the origin of the weighing station. These changes would take several hours to complete and requires a high skill level. The ramp-up phase will also be required after the changes have been made to ensure the ETS is functioning correctly.

## **5.5 Change in Pallet**

An experiment was conducted to examine the effects that changes of the pallet have on the ETS. The In-pallet was moved to a new position on the conveyor and the pallet was rotated 90 degrees clockwise. By rotating the pallet the robot has to approach the pallet from a new direction to pick the breakers up successfully. The workspace and origin of the In-pallet was changed in the robot controller as described in Section 5.3. After the ramp-up phase was completed the ETS tested 50 breakers to confirm it was working as intended.

The experiment was conducted successfully and the ETS tested the 50 breakers using the changed In-pallet workspace.

An additional assessment was done to examine what changes are necessary in the ETS when the pitch of the fixtures on the pallet is changed. To accommodate a change in pitch, the mechanical end stops of the gripper have to be changed by an operator. The mechanical end stops of the gripper control the pitch of the gripper

as seen in Figure 11. The new pitch of the gripper needs to be saved in the supervisor holon as the pitch of the gripper is used to pair breakers. The changes can be made in 15 minutes, but would require a medium skill level. The skill level is required because the adjustments need to be precise and a ramp-up phase is required to ensure the robot can pick and place the breakers correctly. This ramp-up phase is similar to the one described in Section 5.3.

## **5.6 Change in Gripper**

An assessment was made to examine the effects that a change in the number of gripping positions has on the ETS. This assessed the scalability of the system. Only the supervisor holon and the robot controller have to be updated to accommodate the change in the number of gripping positions: The robot controller has to be updated with the new tool center point, while the gripper pitch and number of gripping positions variables have to be updated in the supervisor holon. The changes in the software can be done in approximately 15 minutes, but a medium skill level is required. The new gripper would require thorough testing during the ramp-up phase which can take more than an hour. The ramp-up time of this configuration depends on the design of the gripper and how it is attached to the robot.

## **5.7 Manual Override**

An assessment was done to test the manual override capabilities of the ETS. An experiment could not be conducted to test the manual override capabilities of the test racks because the RWT's are still under development. The manual override would be done by using a HMI connected to the ETS's controller. When the manual override is activated the robot would move to a default position that would enable operators to use the test racks. Since each test rack has a PLC that manages the RWTs of that test rack, the RWTs can still be used. The settings of the RWTs can be set by the operators using the HMI on each test rack.

## **5.8 Robustness**

To test the disturbance handling capabilities of the ETS, the following experiments were conducted: An experiment was conducted to test how the control system reacts to a circuit breaker failing the electrical test and another experiment tested how the control system reacts to a RWT failing.

For the first experiment, one of the testers was programmed to fail all the breakers that it tests. This was done to test whether the ETS scraps the correct breaker.

The second experiment required one of the RWTs to fail. The tester holon responsible for one of the test racks was used to simulate the failing of a RWT.

Both the experiments were successfully completed and both experiments tested 100 breakers. The first experiment would scrap the correct breaker every time the robot picked up a breaker that has to be scrapped. During the second experiment the control system did not make use of the faulty tester and placed the breakers correctly. These experiments prove the ETS's ability to successfully handle some disturbances.

## 5.9 Cycle Time

To test the cycle time of the ETS, two test racks were required. No circuit breakers were used during the test since there was not enough test slots built to simulate 16 RWTs. Two test racks were used to simulate the 16 RWTs, the mock-up test rack and a virtual test rack. The virtual test rack was added to the left (Figure 19) of the mock-up test rack. The robot was run at 100% speed and the time the robot takes to place breakers on the Out-pallet was recorded. Video footage was used to determine the time taken by the robot to complete commands that were sent to it. These results were averaged and used to determine the cycle time. Two experiments were conducted with both experiments testing 200 circuit breakers each.

For the first experiment that was conducted, the robot holon sent one command at a time to the robot controller. This means that to pick a breaker up from the In-pallet, the robot holon would first send a move command to the robot controller to move the robot to the In-pallet. Once the robot has finished the command the robot holon would send a command to pick the breaker up. The robot stops after each command which results in a small pause between commands. This affects cycle time adversely. The individual commands can be seen in Table 5 with the time each command takes to complete.

The first experiment was conducted successfully with the ETS testing 200 circuit breakers. The average cycle time was 9.6 seconds with 2 breakers leaving the ETS in each cycle, with very small variations between breakers. This results in a cycle time of 4.8 seconds per breaker and 12.5 breakers per minute.

For the second experiment that was conducted, the robot holon sent combined commands to the robot controller. This was done to make use of the advance run feature of the robot controller. The advance run feature allows the robot to move through several points without stopping. A move-and-place command and a move-and-pick command were created in the robot holon and robot controller to make use of the advance run feature. If the robot has to pick a breaker up from the In-pallet, the robot holon only sends the move-and-pick command. The robot then moves to the In-pallet and picks up the breaker without stopping in-between. The combined commands can be seen in Table 6 with the time each command takes to complete.

The second experiment was conducted successfully with the ETS testing 200 circuit breakers. The average cycle time was 7.5 seconds per cycle for the 200 breakers tested, with two breakers leaving the cell in each cycle. This results in a cycle time of 3.75 seconds per breaker or 16 breakers per minute.

The results show that the combined commands greatly reduced the cycle time of the ETS. The results also show that the movement between the test racks and the pallets account for a large portion of the cycle time. This time can be reduced by using a faster robot and by moving the test racks closer to the conveyor.

**Table 5: Cycle time breakdown of first experiment**

<b>Robot command</b>	<b>Time to complete (seconds)</b>
Move to pick-up pallet	0.8
Pick-up from In-pallet	1.3
Move to tester slots	1.4
Place in tester	1.3
Move to next tester slots	0.8
Pick-up from tester	1.3
Move to Out-pallet	1.3
Place in Out-pallet	1.4
Move to scrap shoot	1.2
Scrap breaker and move to Out-pallet	1
<b>Total (without scrapping)</b>	<b>9.6</b>

**Table 6: Cycle time breakdown of second experiment**

<b>Robot command</b>	<b>Time to complete (seconds)</b>
Move and pick up from In-pallet	1.4
Move and place in tester	2.1
Move and pick up from tester	1.6
Move and place in Out-pallet	2.4
Move, scrap, move and place in Out-pallet	3.4
<b>Total (without scrapping)</b>	<b>7.5</b>

## 5.10 Control Evaluation

In this section the control system of the ETS that was developed and used in this thesis is assessed. The control system will be evaluated in the context of a station controller. The advantages and disadvantages of the developed control system are discussed in this section. The control system will also be compared to the conventional approach to creating an RMS, which is ABC.

The developed control system uses an embedded PC which is readily available and is supported by the OEMs for use in industrial environments. ABC can also use industrial hardware such as industrial PCs, but industrial PCs cost significantly more than embedded PCs.

The developed control system was created using the IEC 61131-3 programming languages and the programming software provided by the OEM. The IEC 61131-3 languages are the industrial standard and are used in most PLCs, therefore the maintenance crews of most automated factories are familiar with the languages. The maintenance crews would only need training on how the control system works to be able to make changes to the software. In contrast, ABC relies on Java or other programming languages for its control system and uses non-OEM development environments to write the code. The level of programming required for ABC exceeds the abilities of most maintenance crews.

In the context of station control, the control system's ability to interface with hardware, such as pneumatic cylinders and sensors, in real-time is critical. Because the developed control system uses an industrial controller (embedded PC) and the IEC 61131-3 programming language, the control system has full control over the inputs and outputs of the industrial controller. The inputs and outputs of the industrial controller are determined by the IO cards connected to the industrial controller (Beckhoff, 2013). The inputs and outputs of the industrial controller can be set and read in real-time directly by the control system. The control system can set bits directly on Profibus and open a gripper by using digital IO if a Profibus card and a digital IO card are connected to the controller. ABC relies on additional software layers to access IO and is therefore not capable of real-time interfacing with hardware. These additional software layers also add complexity to an ABC system.

Due to limitations of IEC 61131-3, holons cannot be instantiated dynamically. This forces the developed control system to instantiate all holons when the program starts. Although the lack of dynamic instances can be viewed as a disadvantage in comparison to ABC where holons can be instantiated dynamically, it can be argued that the requirements and design of the ETS removes the need for dynamic instances. The only holons in the ETS that would require dynamic instantiation are task holons. One of major requirements of the ETS is a fast cycle time. To minimise the cycle time, multiple breakers have to be transported by the robot at the same time and a control approach has to be used

that will optimise the order in which the breakers are transported, as well as where they are transported to. Since each breaker is represented by a task holon, a method had to be devised that will group the task holons so that the robot can move the breakers together. The grouping would also affect the order in which the breakers are moved as well as where they are moved to. To control the order in which the task holons drive production, the order in which the task holons are allowed to communicate with the operational holons of the system is determined by an optimising entity. Therefore the task holons are passive until the optimising entity allows them to communicate. This means that only one task holon is communicating (active) at any given time. Therefore dynamic instances of task holons is unnecessary as the task holons would be passive until they are allowed to communicate.

The ADACOR control approach was used in the developed control system because it uses a supervisor holon that optimises the task holons. The supervisor holon is described in detail in Section 4.5. Because the developed control system relies on the supervisor holon to group and optimise the task holons, the control system becomes dependant on the supervisor holon. All communications to the cell controller also go through the supervisor holon. This means that if the supervisor holon encounters an error and fails, the control system would also fail. In such a situation the ETS would be switched over to manual override and operators would continue testing the circuit breakers. Although the control system can be changed to run without the supervisor holon, the cycle time would be significantly worse than what it would be during a manual override as the holons would not move together or move to the optimal locations.

A supervisor holon would almost always be necessary for a reconfigurable control system of a station controller as industry always strives for lower cycle times. In these cases, the task holons would again be passive until they are allowed to communicate. This means that the developed control approach of this thesis can be applied to most station controllers as dynamic instantiation of task holons would not be necessary.

Some features of the Beckhoff programming software were used that might not be available from other OEMs. These features include the ability to run multiple PLCs on one controller and the linking of variables between the supervisor holon and task holon manager. Further research should be done to investigate how the developed control system can be applied to industrial controllers of other OEM's.

## 6 Conclusions and Recommendations

Due to industry's reluctance to accept the RMS concept, this research was aimed at developing a control system that the industry will be willing to accept. Research is currently being done by the Mechatronics, Automation and Design Research Group that focuses on using other industry accepted technologies to develop RMS in the context of station controllers. Special care was therefore taken in using technologies that have already gained the industry's acceptance.

The Mechatronics Automation and Design Research Group of the University of Stellenbosch is developing a RQA cell for CBI that will improve the consistency of quality of some of their products. The RQA cell will do visual inspections, electrical testing, assembly and labelling of circuit breakers. This cell is being used for various research topics of the research group. The case study for this thesis is the ETS of the RQA cell.

The focus of this thesis was to evaluate a Beckhoff Embedded controller's suitability as a station controller that controls a subsystem in an RMS. The evaluation was done using the control of one of the subsystems of the CBI case study. To maximise the possibility of the industry accepting the developed control system, only IEC 61131-3 was used for the control of the subsystem.

The ETS was designed to be reconfigurable internally using the characteristics of an RMS. A holonic control approach based on the ADACOR architecture was used to create the control system of the ETS. Each holon, with the exception of the task holons, was run in its own thread on the embedded PC.

The requirements of the ETS created unique challenges for the control system as task holons had to be grouped and moved together. The control system uses a supervisor holon to determine which task holons to group. The supervisor holon also determines the best positions to move the breakers to and optimises the order in which the breakers are moved. A task holon manager was created to manage all the task holons in the system, ensuring that the task holons move together once grouped. The functionality of all the task holons were incorporated into the task holon manager using a data table. The task holon manager can successfully simulate the functionality of all of the task holons because only one task holon can be active at any given time. This allows the control system to function without the ability to instantiate holons dynamically.

From the experiments and assessments presented in Chapter 5, the developed control system met all the requirements listed in Section 3.1.2, with the exception of cycle time. The cycle time can somewhat be lowered to meet the requirements by using a faster robot and a gripper with three pickup positions, but one ETS is unlikely to achieve the 1 second cycle time target, unless the space required for



manual operations is eliminated. The manual override capabilities of the ETS could, however, not be tested, because the RWTs were still under development.

The evaluation of the developed control system in Section 5.10 proved that an RMS can be created, in the context of station control, using IEC 61131-3 and industry accepted technologies if a hardware platform is used that allows multiple PLCs to be run in individual threads. The control approach that was created in this thesis can be used to create station based control systems that offers optimised cycle times and the benefits of an RMS in combination with benefits of industry accepted technology.

Further research can be done in using a Beckhoff Embedded PC to create RMSs using the C++ extensions to augment the functionality of the supervisor holon. The C++ extensions can be used to create more advanced optimisation algorithms.

Further research should be done to investigate how the developed control system can be implemented on industrial controllers of other OEMs like Siemens. Companies often have a preferred choice of OEM for the industrial controller they use in their factories. If the control system can be successfully implemented on other OEM's industrial controllers, the industry would be more even more likely to adopt the control approach.

## 7 References

- Adams, A.O., 2010. *Control of reconfigurable assembly system*. MSc Eng Thesis. University of Stellenbosch.
- Albadawi, Z., Boulet, B., DiRaddo, R., Girard, P., Rail, A. & Thomson, V., 2006. Agent based control of manufacturing processes. *International Journal of Manufacturing Research*, 1 (4), pp.466 - 481.
- Almeida, F.L., Terra, B.M., Dias, P.A. & Gonçalves, G.M., 2010. Adoption Issues of Multi-Agent Systems in Manufacturing Industry. In *Fifth International Multi-conference on Computing in the Global Information Technology*, pp.238 – 244.
- Beckhoff, 2013. *TwinCAT3*. [Online] Available at: [www.beckhoff.com](http://www.beckhoff.com) [Accessed 1 June 2013].
- Bosch Rexroth, 2013. *Open Core Engineering*. [Online] Available at: [http://www.boschrexroth.com/en/xc/products/engineering/opencoreengineering/the\\_features\\_of\\_open\\_core\\_engineering/open\\_core\\_interface/open\\_core\\_interface](http://www.boschrexroth.com/en/xc/products/engineering/opencoreengineering/the_features_of_open_core_engineering/open_core_interface/open_core_interface) [Accessed 1 June 2013].
- Brückner, S., Wyns, J., Peeters, P. & Kollingbaum, M., 1998. Designing Agents for Manufacturing Control. In *Proceedings of the 2nd AI & Manufacturing Research Planning Workshop*, pp.40 - 46.
- Chirn, J.-L. & McFarlane, D.C., 2000. A holonic component-based approach to reconfigurable manufacturing control architecture. In *International Workshop on Database and Expert Systems Applications*, pp.219 - 223.
- Christensen, J., 1994. Holonic Manufacturing Systems: Initial architecture and standards directions. *Proceedings of the First European Conference on Holonic Manufacturing Systems, Hanover, Germany*.
- Duffie, N.A. & Piper, R.S., 1986. Nonhierarchical control of manufacturing systems. *Journal of Manufacturing Systems*, 5 (2), p.141.
- Dymond, F.S.D., 2009. *Conceptual design of a fixtureless reconfigurable automated assembly system*. MSc Eng Thesis. Stellenbosch University.

- Eurobots, 2012. [Online] Available at:  
[http://www.eurobots.net/files/schemes/det\\_214\\_Kuka\\_KR15\\_eurobots13.jpg](http://www.eurobots.net/files/schemes/det_214_Kuka_KR15_eurobots13.jpg)  
[Accessed 1 October 2013].
- Heikkilä, T., Jarviluoma, M. & Juntunen, T., 1997. Holonic control for manufacturing systems: design of a manufacturing robot cell. *Integrated Computer Aided Engineering*, 4, pp.202 – 218.
- Hoffman, A.J., Basson, A. & Le Roux, A., 2013. Towards Alternatives for Agent Based Control in Reconfigurable Manufacturing Systems. In *5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, pp.237 - 242.
- Koestler, A., 1967. *The Ghost in the Machine*.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. & Van Brussel, H., 1999. Reconfigurable Manufacturing Systems. *CIRP*, 48 (2), pp.527 - 540.
- Koren, Y. & Shpitalni, M., 2010. Design of reconfigurable manufacturing systems. *Journal of manufacturing systems*, 29, pp.130 - 141.
- Kruger, K., 2013. *Control of the feeder for a reconfigurable assembly system*. MSc Thesis. Stellenbosch University.
- Le Roux, A., 2013. *Control of a conveyor system for a reconfigurable manufacturing cell*. MSc Eng Thesis. Stellenbosch University.
- Leitão, P., 2009. Agent-based distributed manufacturing control: A state of the art survey. *Engineering Applications of Artificial Intelligence*, 22 (7), pp.979 - 991.
- Leitão, P. & Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, (57), pp.121 - 130.
- Liu, S., Gruver, W.A., Kotak, D. & Bardi, S., 2000. Holonic manufacturing system for distributed control of automated guided vehicles. In *2000 IEEE International Conference on Systems, Man and Cybernetics*, pp.1727 – 1732.
- Marik, V., 2005. Industrial adoption of agent-based technologies. *Intelligent Systems IEEE*, 20 (1), pp.27 - 35.

Maturana, F. & Norrie, D., 1996. Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, 7, pp.257 - 270.

Monch, L., Stehli, M. & Zimmermann, J., 2003. FABMAS: an agent-based system for production control of semiconductor manufacturing process. In Marik, V., McFarlane, D. & Valckenaers, P. *Holonic and Multi-Agent Systems for Manufacturing. Lecture Notes in Artificial Intelligence*. Heidelberg: Springer. pp.258 - 267.

Monostori, L. & Kadar, B., 1999. Agent-based control of manufacturing systems. *Proceedings of the Second International Conference on Intelligent Processing and Manufacturing Materials*, 1, pp.131 - 137.

Mulubika, C., 2013. *Comparison of IEC 61499 and agent based control for a reconfigurable manufacturing system*. MSc Eng Thesis. Stellenbosch University.

Van Dyke Parunak, H., Baker, A. & Clark, S., 1998. The AARIA agent architecture: An example of requirements-driven agent-based system design. In *First International Conference on Autonomous Agents*, pp.482 - 483.

Sauter, T. & Massotte, P., 2001. Enhancement of distributed product systems through software agents. In *Proceedings of Eight IEEE International Conference on Emerging Technologies and Factory Automation*, pp.267 - 272.

Sequeira, M.A., 2009. *Conceptual design of a fixture-based reconfigurable spot welding system*. MSc Eng Thesis. University of Stellenbosch.

Siemens, 2013. *WinAC RTX*. [Online] Available at: <http://www.automation.siemens.com/mcms/programmable-logic-controller/en/software-plc/simatic-winac-rtx/Pages/Default.aspx> [Accessed 1 June 2013].

The Foundation for Intelligent Physical Agents, 2012. *FIPA*. [Online] Available at: <http://www.fipa.org/> [Accessed August 2012].

Tönshoff, H.K. & Winkler, M., 1996. Shop control for holonic manufacturing systems. *CIRP Proceedings-Manufacturing Systems*, 20 (3), pp.277 - 281.

Valckenaers, P., Heikkilä, T., Baumgaertel, H., McFarlane, D. & Courtois, J., 1999. Towards a novel manufacturing control principle. In *Proceedings of the Second International Workshop on Intelligent Manufacturing*, pp.871 - 875.

Valckenaers, P. & Van Brussel, H., 2005. Holonic Manufacturing Execution Systems. *CIRP Annals*, 54, pp.427 - 432.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. & Peeters, P., 1998. Reference architecture for holonic manufacturing systems. In *Computers in industry*, 37, pp.255 - 274.

Vrba, P., 2012. Past , Present and Future of Distributed Intelligent Control in Industrial. In *Information Control Problems in Manufacturing*, 14, pp.22 - 25.

Vrba, P. & Marik, V., 2005. Simulation in agent-based control systems: MAST case study. In *Proceedings of the 16th IFAC World Congress*. CD-ROM