OULU 2014
C 481

ACTA UNIVERSITATIS OULUENSIS

Michael Peschl

AN ARCHITECTURE FOR FLEXIBLE MANUFACTURING SYSTEMS BASED ON TASK-DRIVEN AGENTS

UNIVERSITY OF OULU GRADUATE SCHOOL;
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING,
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING;
INFOTECH OULU

ACTA UNIVERSITATIS OULUENSIS
C Technica 481

MICHAEL PESCHL

AN ARCHITECTURE FOR FLEXIBLE MANUFACTURING SYSTEMS BASED ON TASK-DRIVEN AGENTS

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Technology and Natural Sciences of the University of Oulu for public defence in Auditorium TS101, Linnanmaa, on 21 February 2014, at 12 noon

UNIVERSITY OF OULU, OULU 2014

Acta Univ. Oul. C 481, 2014

Supervised by
Professor Juha Röning

Reviewed by
Doctor Niels Lohse
Docent Paul Valckenaers

Opponents
Doctor Philipp Dreiss
Doctor Niels Lohse

ISBN 978-952-62-0365-2 (Paperback)
ISBN 978-952-62-0366-9 (PDF)

ISSN 0355-3213 (Printed)
ISSN 1796-2226 (Online)

Cover Design
Raimo Ahonen

JUVENES PRINT
TAMPERE 2014

Peschl, Michael, An architecture for flexible manufacturing systems based on task-driven agents.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering, Department of Computer Science and Engineering; University of Oulu, Infotech Oulu
Acta Univ. Oul. C 481, 2014
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

During the last decades significant changes in the buying behavior of customers can be observed. While in former days price sensitivity lead to more uniformed products, in present days manifold high-quality products and customization with reasonable prices and rapid delivery are demanded.

As a consequence, the industry asks for manufacturing systems which allow for fast ramp-up, multi-variant production and rapid adaptability. In this environment, several scientific approaches such as agent-based and holonic manufacturing systems have been investigated within the last years.

In order to cover all aspects of the foreseen future demands, the architectures for such systems are very complex and the system's entities are characterized by very flexible behavior. Hence, the efforts for their implementation are rather high and the systems tend to exhibit non-deterministic behavior. Furthermore, the top down approach of most systems leads to a complete re-organization of the factory management. As a consequence the acceptance for such systems in real industrial environment at present day is very limited.

Therefore, the objective of this thesis is to develop an architecture for flexible manufacturing systems which allows for easy take-up in the industry. It is based on a bottom-up approach with a new kind of flexible, intelligent shop-floor components called Manufactrons. The architecture covers all layers of traditional factory organization with special emphasis on the shop floor organization. The approach and results are based on the research activities of the European Research Project XPRESS in which representatives of three major industry branches collaborated in order to find a solution for their future demands on flexible manufacturing systems.

The architecture has been implemented in the context of XPRESS in aerospace, automotive and electrical industry. The tests show the feasibility of the approach. The capability for a smooth integration of the new approach into existing manufacturing environment has successfully been demonstrated.

Keywords: flexible, industrial application, Manufactron, manufacturing, shop-floor organization, task-driven

Peschl, Michael, Tehtäväohjattuihin agentteihin perustuva arkkitehtuuri mukautuvia tuotantojärjestelmiä varten.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta, Tietotekniikan osasto; Infotech Oulu
Acta Univ. Oul. C 481, 2014
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Viime vuosikymmeninä asiakkaiden ostokäyttäytyminen on muuttunut merkittävästi. Ennen asiakkaiden hintatietoisuus johti yhtenäisiin tuotteisiin, kun taas nykyään vaaditaan moninaisempia tuotteita ja muokattavuutta kohtuulliseen hintaan. Samaan aikaan odotetaan korkealaatuisia tuotteita ja nopeaa toimitusta. Nämä seikat ovat aiheuttaneet tuotantoteollisuudelle uusia haasteita. Reagoidakseen nopeasti asiakkaiden vaatimuksiin tuotannonsuunnittelussa on alettu keskittymään korkealaatuisten tuotemuunnelmien määrän kasvattamiseen.

Tämän vuoksi tarvitaan tuotantojärjestelmiä, jotka mahdollistavat nopean Ramp Up -prosessin, moneenmuuntuvan tuotannon ja nopean mukautuvuuden. Tätä aihetta on viime vuosina lähestytty esimerkiksi agentteihin perustuvien ja holonisten tuotantojärjestelmien kautta. Kuitenkin näihin tulevaisuuden haasteisiin pystytään vastaamaan vain kompleksisilla arkkitehtuureilla ja järjestelmän entiteeteille ominaisia ovat hyvin mukautuvat käyttäytymismallit. Näiden toteuttamiseen tarvitaan paljon työtä ja järjestelmillä on tapana käyttäytyä epä-deterministisesti. Lisäksi ylhäältä alas lähestymistapa johtaa usein tehtaan täydelliseen uudelleenorganisointiin, minkä vuoksi lähestymistapaa ei suosita oikeissa teollisuusympäristöissä.

Tämän väitöstyön tarkoituksena on kehittää mukautuville tuotantojärjestelmille arkkitehtuuri, joka mahdollistaa järjestelmien helpon käyttöönoton teollisuudessa. Arkkitehtuuri perustuu alhaalta ylös -lähestymistapaan ja sisältää uudenlaisen joustavan ja älykkään tuotantotilakomponentin, manufactronin. Arkkitehtuuri kattaa kaikki perinteisen tehdasorganisaation kerrokset keskittyen kuitenkin erityisesti tuotantotilojen organisointiin.

Lähestymistapa ja tulokset perustuvat Euroopan Unionin XPRESS-tutkimusprojektiin. Projektissa tehtiin yhteistyötä kolmen suuren teollisuushaaran kanssa tarkoituksena löytää joustava tuotantojärjestelmäratkaisu tulevaisuutta varten. Arkkitehtuuria sovellettiin XPRESS-projektissa lentokone-, auto- ja elektoniikkateollisuuteen ja testit osoittivat lähestymistavan soveltuvuuden. Myös lähestymistavan sujuva integrointi olemassa oleviin teollisuusjärjestelmiin osoitettiin onnistuneesti.

Asiasanat: joustava, Manufactron, tehtäväohjattu, teollisuussovellus, tuotanto, tuotantotilojen organisaatio

Für meine Oma
Julie Peschl

Preface

> *”Some people say that there are no more frontiers left for us to conquer. But then again, some people still go out into the wilderness in search of their dream.”*
>
> *– MacGyver*

This thesis is the result of a long way of a fascinating idea. At the very beginning there was not more than the long held dream of my former boss at Harms & Wende to finally develop a welding control unit which needs no sophisticated setup programming, but works properly only by telling something like: “*Weld two steel sheets with 1.5mm thickness and produce a spot diameter of 4mm*”. While thinking about this vision, I realized the potential of it, not only for welding controls, but for an entire manufacturing system – the task-driven manufacturing paradigm was born. Several attempts for getting funding for the idea finally ended in the European Research Project XPRESS that I was in charge of as the overall project coordinator.

Now while I am writing the last few pages I realize how many people contributed to the success of this work.

My first gratitude goes to the entire XPRESS consortium. For 4 ½ years we had a fruitful working atmosphere, wonderful cooperation and friendship. Without your great initiative and enthusiasm for the project it would not have been possible to make all the things working. Thank you, guys!

Very special thanks go to my supervisor Dr. Juha Röning for giving me the opportunity to carry out this thesis at the University Oulu and for his excellent scientific support. I deeply honor that he accepted me without any hesitation as a PhD student and that he settled all the organizational issues for me which are unavoidable when doing the research work more than 2,000 km away from the University.

I am very grateful to the reviewers of this thesis, Dr. Niels Lohse and Dr. Paul Valckenaers. Their comments helped very much to focus this thesis on the essentials and significantly improved its quality. It was a privilege to have such experts for the review process.

In addition, I would also like to express my gratitude to Dr. Sabine Preusse for her tireless patience in correcting my thesis and her ideas for improving this work.

Many thanks also go to my colleagues from Harms & Wende and in particular to my team in Karlsruhe who are doing an excellent job. It is a pleasure to work with you!

I also would like to thank Dr. Norbert Link from the University of Applied Science Karlsruhe, Germany for a long-lasting great cooperation in various research activities and especially for his never-ending wealth of “generic” ideas.

In addition I like to thank the office members of the Institute of Applied Research for sharing breakfast and lunch with the “Harms & Wende aliens” in the university’s premises. Very special thanks go to Corinna Kloiber for her laughter and for a wonderful friendship.

The deepest and warmest gratitude belongs to my family: My uncle Jürgen who is definitely the one who inspired my fascination with technology and computer science; my sister Birgit for being my sister and for giving me together with her partner Andreas my nephews Kian “Hallalüll” and Levin – the guys who are the future of our family; finally my mom Ursula for her faith and her encouragement throughout all my activities. Thanks to everybody for their constant support and belief in me and my dreams.

Mühlacker, 30th December 2013 Michael Peschl

List of abbreviations

AARIA	Autonomous Agents for Rock Island Arsenal
ABAS	Actor Based Assembly System
ACL	Agent Communication Language
ADACOR	Adaptive Holonic Control Architecture for Distributed Manufacturing Systems
AGV	Automated Guided Vehicle
APC	Advanced Process Control
API	Application Programming Interface
APS	Advanced Process Simulation
CAD	Computer-Aided Design
CFRP	Carbon-Fiber-Reinforced Plastic
CICA	Component-based Intelligent Control Architecture
DDL	Device Description Language
DDM	Direct-Digital Manufacturing
e.g.	exempli gratia
eBOP	electronic Bill Of Processes
EDDL	Electronic Device Description Language
ERP	Enterprise Resource Planning
EU	European Union
FC	Factory Co-ordination
FEM	Finite Element Method
FIPA	Foundation for Intelligent Physical Agents
FMS	Fractal Manufacturing Systems
GDP	Gross Domestic Product
GLARE	GLAss fibre REinforced aluminium
GPOS	General Purpose Operating System
HMS	Holonic Manufacturing Systems
ICT	Information and Communication Technologies
IEC	International Electrotechnical Commission
IMS	Intelligent Manufacturing Systems
IT	Information Technologies
JADE	Java Agent Development Framework
KPI	Key Performance Indicator
KQML	Knowledge Query and Manipulation Language
KRL	KUKA Robot Language

MAS	Multi Agent System
MES	Manufacturing Execution System
MSI	Manufacturing Systems Integration
NC	Numerical Control
OEM	Original Equipment Manufacturer
OLE	Object Linking and Embedding
OOP	Object-Oriented Programming
OPC	OLE for Process Control
P2P	Production 2000+
PAC	Production Activity Control
PES	Production Execution System
PFS	Production Feedback System
PLC	Programmable Logic Controller
PROSA	Product-Resource-Order-Staff Architecture
QRD	Quality Result Document
R&D	Research&Development
REQ	Requirement
REST	Representational State Transfer
RFID	Radio-Frequency Identification
RP	Rapid Prototyping
RT	Rapid Tooling
SCARA	Selective Compliance Assembly Robot Arm
SL	Semantic Language
SPC	Statistical Process Control
TDD	Task Description Document
U.S.	United States
W3C	World wide web consortium
WES	Workflow Execution System
WF	Windows Workflow Foundation
WfM	Workflow Manager
WPF	Windows Presentation Foundation
XML	EXtensible Markup Language
XPRESS	FleXible PRoduction Experts for reconfigurable aSSembly technology

Contents

1 Introduction

This chapter provides a general introduction of the thesis. First, the background and environment of the thesis is given. After that, the problem description and motivation for the design of a novel manufacturing control architecture is discussed. Then, the overall goal and the contributions of the thesis are described. The introduction ends with an illustration of the thesis outline.

1.1 Background of the thesis

The manufacturing industry is one of the most important drivers for the worldwide economy. In 2010, manufacturing gained more than 16% of the total gross domestic product (GDP) (World Bank 2013). In the United States, manufacturing employed 13 million workers which generated $1.6 trillion to the GDP in 2009 (Tassey 2010). The EU-27 manufacturing sector employed 34 million people in 2006 which was equivalent to 27 % of the employment in the non-financial business economy (Sura 2009). To ensure the worldwide competitiveness of the manufacturing industry, enormous efforts and investments are done. For example, in the United States, almost three-fourths of R&D is in the manufacturing sector (Tassey 2010). While Europe and the United States are still very strong in their efforts on R&D in manufacturing, the countries with the highest growth rates of investments in R&D are China (about 20%), India (about 12%) and South Korea (about 12%) (Shipp *et al.* 2012). Besides the R&D efforts, manufacturing has also a significant share in the expenditures of the industry. According to Smith (2006) the manufacturing operations ranked at position 1 of the company's expenditures, followed by Enterprise Resource Planning (ERP) related positions. More in detail, the highest investments in manufacturing operations have been projected in Manufacturing Execution Systems (MES), followed by product and process quality management, advanced process control and simulation (APC/APS) and product data and/or recipe management.

1.2 Problem description and motivation

During the last decades significant changes in the buying behavior of customers can be observed. While in former days price sensitive buying behavior led to more standardized and uniformed products, in present days manifold products and product customization with reasonable prices are demanded. At the same time, high product quality and rapid delivery is expected. Under these circumstances the manufacturing industry is faced with various challenges. Fast reaction to customers' demands, an increasing number of product variants and high-quality production is in the focus of production planning. In addition to that, further aspects such as product and production sustainability or environmental friendliness by increased energy efficiency or a reduced carbon dioxide emission have to be taken into account.

As a consequence, flexible and extendable manufacturing systems are required which allow for fast ramp-up of production, multi-variant production and rapid production adaptability. In order to realize these features, efficient production planning tools and new production ramp-up strategies are needed. Furthermore, highly flexible and adaptable production systems are needed in order to produce of multiple products and variants on the same production system, having also the capability of dynamic reaction on changed boundary conditions of producing such as optimization goals or requested quality standards. On shop-floor, intelligent, self-adaptive and modular devices are required, which can be brought to operation by just inserting them into the production network. In addition to that, the smooth integration of workers and operators in an automated environment is also required, as humans are the most flexible resource available. In this environment, high efforts in R&D for the future generation of manufacturing systems are done. Strategic visions and roadmaps are created in which the most important steps towards innovative solutions for adaptive manufacturing are manifested.

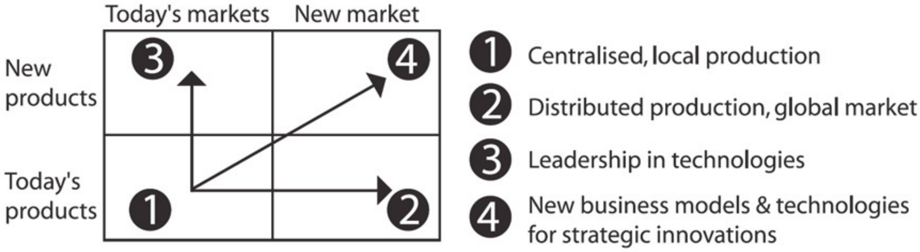


Fig. 1. MANU*FUTURE* scenarios on today's and new markets and products (MANUFUTURE 2004, published by permission of the European Commission).

The European Organization Manu*Future* for example identified the distributed production on a global market with technological leadership and new business models as the most important trends in future manufacturing (Fig. 1). The transition from resource-based to knowledge-based manufacturing is seen as one of the most important success factors. (MANUFUTURE 2004).

Another example for future manufacturing is the concept of the smart factories proposed by the U.S. non-profit organization Smart Manufacturing Leadership Coalition. In their vision smart factories are built of knowledge-based applications for production optimization and cost reduction. Smart factories are relying on interoperable systems such as flexible, IT-enabled supply chains, links between product development/design and the manufacturing process as well as on real-time information flow between factories and distribution centers. Customized products shall be produced which are traced throughout their service lifetime and which can be recycled or remanufactured. (SMLC 2011).

On the basis of such general roadmaps, concrete specifications and implementations for future manufacturing systems have been derived. In order to cover all relevant aspects of the foreseen future demands on manufacturing systems, the architectures for such systems are very ambitious and complex and the system's entities are characterized by very flexible behavior. Hence, the efforts for their implementation are rather high and the systems tend exhibit non-deterministic behavior. Furthermore, the top down approach of most system implementations leads to the need of complete re-organization of the factory management. As a consequence, the acceptance for such systems in real industrial environment at present days is very limited. For that reason it is required to have a more detailed view on the specific needs of the industry for flexible manufacturing systems and to derive a system architecture which on the one hand includes all relevant elements for flexible and adaptable production, but on the other hand takes all specific needs and potential limitations given from the industry into account.

1.3 Objectives and contribution

The objective of this thesis is to develop an architecture of flexible manufacturing systems which allows for easy take-up in the industry. It is based on a *bottom-up approach* with a new kind of flexible, intelligent shop-floor components called *Manufactrons*. The approach and results are based on the authors' research and development activities related to the European Research Project XPRESS (XPRESS 2011). As part of XPRESS, representatives of the three major industry branches Aerospace, Automotive and Electrical industry as well as the related system integrators and component suppliers collaborated in order to find a proper solution for their future demands on flexible manufacturing systems. Demand is seen in the following fields:

- Decrease production ramp-up time;
- Provide an easy to use methodology for multi-variant production;
- Allow for an easy implementation in the industry;
- Generate and exploit knowhow of component and system vendors;
- Comprehensive documentation of production data.

Prior to the project execution the author of this thesis was responsible for gathering the first basic industry requirements and for deriving the basic system architecture. Those activities resulted in the preparation of the project proposal for which the author was the main driver. As part of the XPRESS project, the author was the overall project coordinator and as the leader of the so-called Scientific Coordination Team responsible for defining and monitoring the general scientific objectives and progress of the project. Against this background, the specific contribution of the author in the context of XPRESS was to develop the basic concepts of the system architecture, namely i.) the principles of task-driven manufacturing, ii.) the hierarchical organization and communication models of production entities and iii.) the orchestration and quality documentation of production. In addition to that, the author also developed the concepts for the integration of humans in a task-driven environment, the approaches for distributed simulation and emulation and the interfacing of ERP systems for workflow generation. Furthermore, the author accompanied the requirement gathering process at the different industrial partners and was responsible for the alignment of the consolidated overall requirements with the project scope and vice versa. During the implementation and test phase of the project the author concentrated on the definition and implementation of the Welding Manufactron. These activities also included the definition of the Manufactrons' self-description, the related Task Description Documents und Quality Result Document as well as the implementation of the interface with a real-time system for welding processes. Furthermore, a sample implementation of a Super Manufactron for the cooperation of the Welding Manufactron with other Manufactrons was done by the author.

In light of these contributions to the project, the focus of the work is set on the *design of the theoretical framework* of the overall architecture[1]. Technical details are provided partially for further clarification.

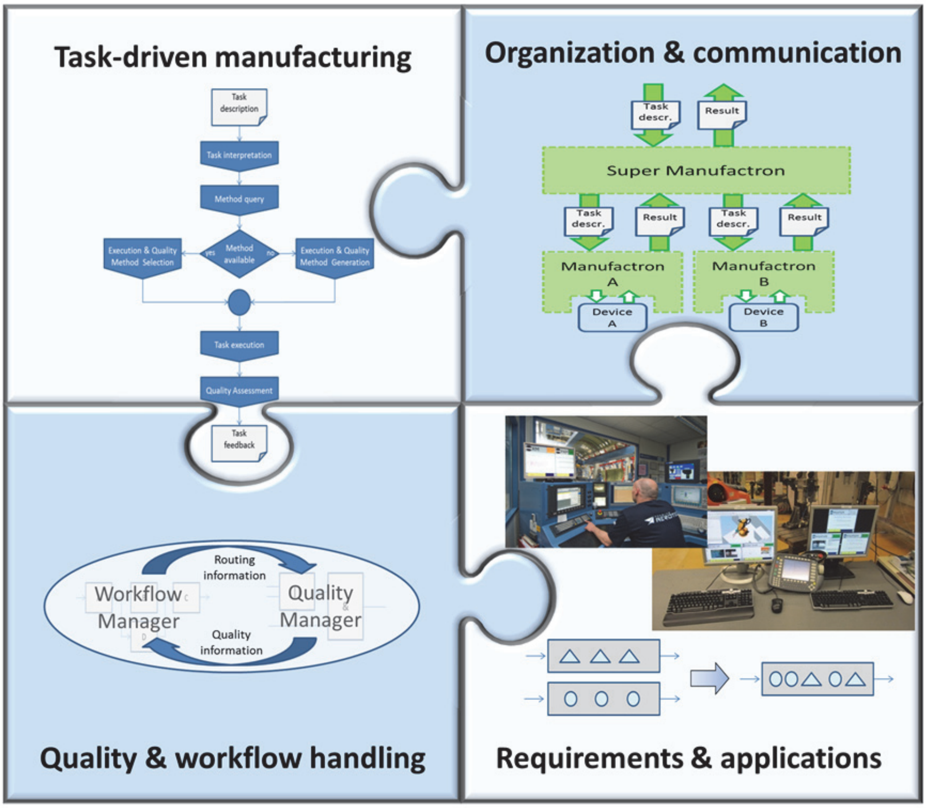


Fig. 2. The four major subjects of the thesis: (1) Task-driven manufacturing; (2) Organization and communication of and between Manufactrons; (3) Quality and workflow handling and (4) Requirements and applications.

As depicted in Fig. 2 the thesis consists of four major research subjects:

The first subject focuses on *task-driven manufacturing,* defining the principles of task-driven manufacturing and the internal structure of task-driven devices. Furthermore, a definition of the Manufactrons from different perspectives is provided.

---

[1] Parts of the architecture have also been reported in Peschl&Hoffmeister (2011) and in Peschl *et al.* (2011).

The second subject defines the *organization* of Manufactrons at shop-floor level using Super Manufactrons and the *communication* between Manufactrons and other production entities, including mechanisms for task description and quality result communication, event and synchronization mechanisms.

The third subject defines the *workflow and quality handling* at MES level. To this end, Workflow Managers and Quality Managers are introduced which are supposed to transport task-relevant data to the Manufactrons and to gather and assess quality information from Manufactrons.

The fourth subject is related to the *industrial acceptance and implementation* of the architecture. It consists of a description of the requirement gathering and assessment process for several relevant industry branches as well as the illustration of different applications providing a proof of concept.

1.4 Outline of the thesis

This thesis is structured in six chapters as illustrated in Fig. 3.

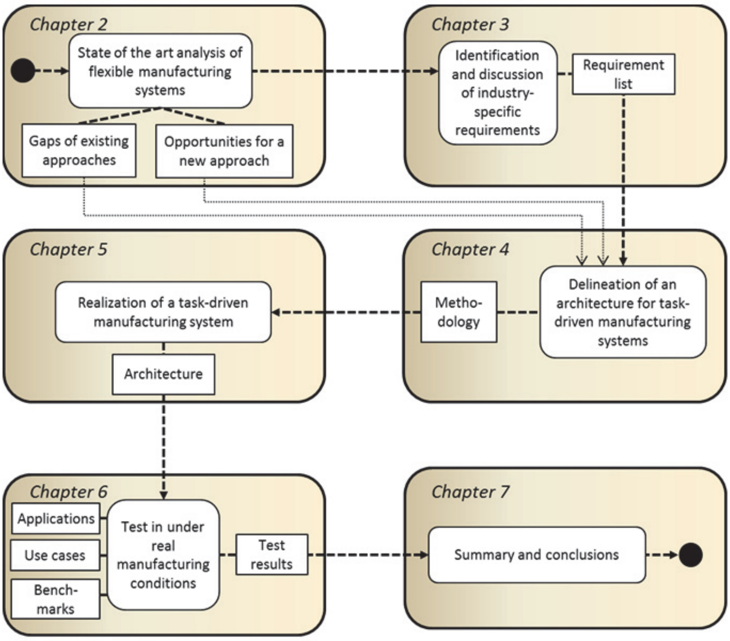


Fig. 3. Structure of the thesis, covering the main chapters: State of the art analysis, requirements identification, architecture delineation, realization and validation.

In Chapter 2, the current state of the art for flexible manufacturing systems is discussed. Several approaches for manufacturing control architectures and the latest research in this area are described. The focus is set to the readiness of the approaches for industrial implementation. The discussion ends with the identification of the barriers for a successful implementation of the approaches in the industry. Chapter 2 ends with the identification of the major research questions of the thesis and with a description of the underlying research methodology.

In Chapter 3 the requirements for a novel manufacturing system architecture which is applicable for the industry branches under consideration are described. The outcome of Chapter 3 is a set of ten specific requirements which need to be fulfilled for a successful implementation of a flexible manufacturing in the respective industries.

In Chapter 4 a delineation of an architecture for task-driven manufacturing systems is provided. This chapter gives an overview of the basic concepts of the architecture. Chapter 4 draws a whole picture of the overall aspects of a task-driven manufacturing system and enables the reader to understand the general idea. The outcome of Chapter 4 is the methodology for such a system.

The methodology is used as an input for Chapter 5. In this chapter the three main pillars of a task-driven manufacturing system architecture are described in detail. Each pillar is illustrated in a dedicated section which ends with a discussion by reflecting the requirements identified in Chapter 3. At the end of Chapter 5 the reader has a complete view on the architecture of the novel approach of a task-driven manufacturing system.

In Chapter 6 the approach is validated. To do so, various application examples from different industry branches are described. On the basis of use-cases and benchmarks the feasibility of the approach is demonstrated.

This thesis ends with Chapter 7 in which a summary and the conclusions on the architecture are provided.

2 Scientific state of the art of manufacturing control architectures

In this chapter the scientific and technological state of the art of flexible and adaptive manufacturing systems is described. Of special interest are those approaches which have already been implemented in an industrial environment or at least in a laboratory test bed. Special attention will be on the Multi-agent Systems (MAS) and the Holonic Manufacturing Systems (HMS) with special emphasis on their capabilities for embedding of the related methodologies in the factory structure and on the handling of knowledge and expertise within the various domains.

2.1 Traditional control architectures

During the last decades several approaches for control architectures of manufacturing systems have been developed. Commonly, centralized, hierarchical, modified hierarchical and heterarchical systems are distinguished (Dilts *et al.* 1991). Fig. 4 provides an overview of the various organization approaches.

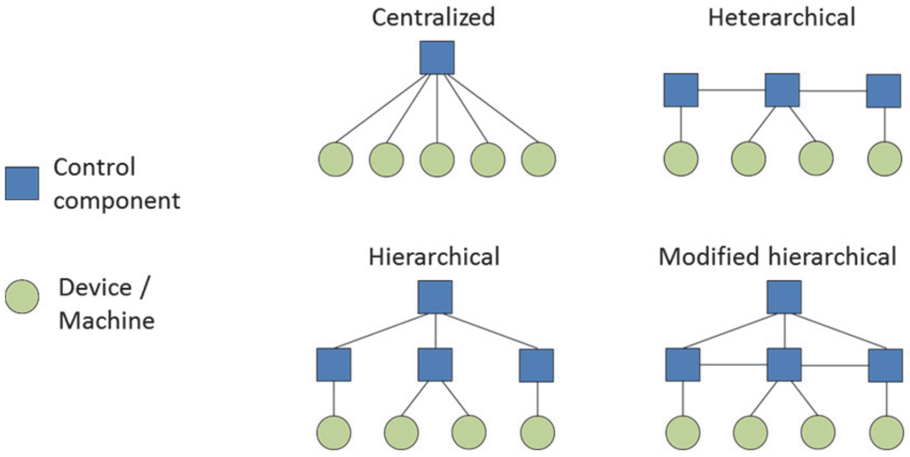


Fig. 4. Control architectures of manufacturing systems (cf. Dilts *et al.* 1991, published by permission of © Elsevier).

The evolution of control architectures started with centralized control systems. Such systems are characterized by a single unit which decides on the control of the underlying machines and devices (Dilts *et al.* 1991). The major advantages of such systems are the global access to relevant information and data, the possibility for easy global optimization and the avoidance of redundancies in data storage. However, on the other hand, centralized architectures are also characterized by strong disadvantages. The high complexity of the central control unit, the bad response time of the machines and devices and the overall very limited fault tolerance (breakdown of the control instance) of the whole system are the major negative aspects.

The hierarchical organization model is characterized by the decomposition of one complex structure into several smaller entities. The information flow is done by sending commands top-down and by providing feedback bottom-up. The higher the entity resides in the hierarchy, the more abstract the commands are which are issued and received. A prominent example of a hierarchical control architecture is the Production Activity Control (PAC) architecture which describes the building blocks and communication within a production cell on cell level (Browne 1988). PAC modules are coordinated by Factory Co-ordination modules (FC), which reside on the factory level (Bauer 1994). The main advantages of hierarchical control are their robustness, fast response times, predictable system behavior and the reduced implementation complexity. Contrarily, the presence of various communications links and the poor system behavior in case of disturbances reduce the system performance. The fairly static structure hampers the modification of the structure e.g. if an entity has to be exchanged or if an alternative process plan has to be executed (see also Duffie&Prabhu 1994).

The modified hierarchical architecture extends the classical hierarchical architecture by providing a peer-to-peer communication of the control entities. Prominent examples of such architectures are PAC++ (Martensson *et al.* 1997), RapidCIM (Wysk 2009), and MSI (Senehi 1994). The peer-to-peer communication allows for better synchronization, for fast data communication and for an improved disturbance handling. In addition to that, local autonomy can be realized. However, most of the negative aspects of the classical hierarchies still remain.

In order to overcome the limitations of hierarchical architectures, especially the inflexibility for adaptations and rapid changes during production, flat organizational structures have been designed. Such heterarchical organization structures are characterized by a horizontal flow of information. All production entities reside on the same organization level. First concepts of such systems have been introduced by Hatvany (1985). Further research work has been done for example by Duffie&Prabhu (1994) who developed a distributed scheduling method for decentralized systems and by Veeramani *et al.* (1993) who investigated heterarchical organization structures for large flexible manufacturing systems. Both authors identified the reduced complexity, fault-tolerance and adaptability as the main advantages of heterarchical organization structures. However, the missing supervising entity increases the likelihood of optimizing the system only locally and also of non-deterministic and unpredictable system behavior.

2.2 Agent-based control architectures

As a consequence of the disadvantages of centralized and hierarchical systems, new approaches of completely decentralized system architectures have been investigated. With the explosive growth of technologies for computers, communication and information exchange, the basis for agent-based computation has been served. Agent-based software systems have been seen as a new paradigm in computation and software development. As agents are characterized by autonomy, self-responsibility and self-recovery, such technologies address also the major aspects of (future) manufacturing systems (van Dyke Parunak 1998). Russel *et al.* (2010) distinguish four different agent types: *Simple reflex agents* which select actions only on the basis of the current percept without taking account of the percept history; *Model-based reflex agents* which include a model of its environment and keeping track of its changes; *Goal-based reflex agents* which enrich the models by goals that describe desirable situations and finally *Utility-based agents* which are adding degrees of usefulness to the goals. Learning elements as a part of the agent can improve their performance.

Despite the fact that agents and agent-based systems do not have one common definition, a general consensus about two main abstractions exists (Monostori *et al.* 2006):

- An *agent* is a computational system that is situated in a dynamic environment and is capable of exhibiting autonomous and intelligent behavior.
- An agent may have an environment that includes other agents. The community of interacting agents, as a whole, operates as a *multi-agent system* (*MAS*).

According to Monostori *et al.* (2006) agents are characterized by the following key properties:

- Agents act on behalf of their designer or the user they represent in order to meet a particular purpose;
- Agents are autonomous in the sense that they control both their internal state and behavior in the environment;
- Agents exhibit some kind of intelligence, from applying fixed rules to reasoning, planning and learning capabilities;
- Agents interact with their environment, and in a community, with other agents;
- Agents are ideally adaptive, i.e. capable of tailoring their behavior to the changes of the environment without the intervention of their designer.

A multi-agent system is built by a network of agents which are interacting and communicating in order to reach common goals (Fig. 5).

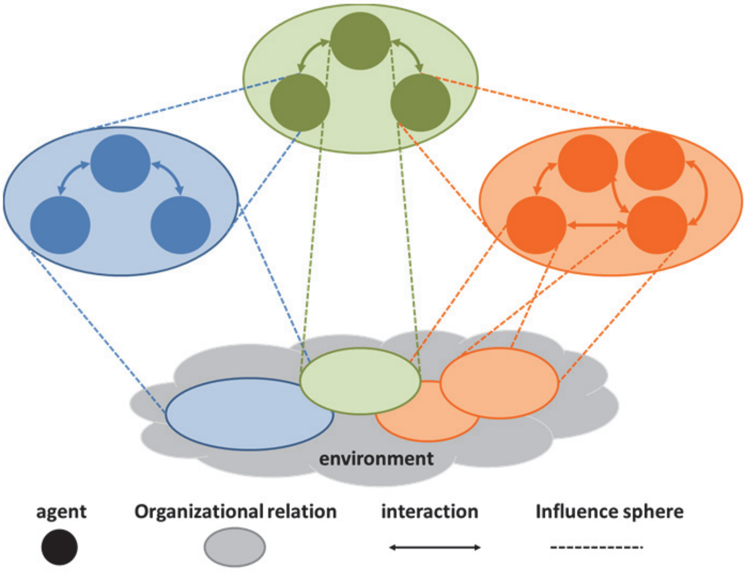


Fig. 5. General scheme of a multi agent system (Jennings 2001, published by permission of © ACM).

To do so an agent communication language (ACL) is required. Prominent examples are the Knowledge Query and Manipulation Language (KQML) and the ACL of the Foundation for Intelligent Physical Agents (FIPA ACL). KQML consists of 3 layers, namely the content layer, the message layer and the communication layer (Finin *et al.* 1994). In the content layer the actual content of the message is represented. The message layer is used for the message transport between applications. Finally, the communication layer consists of low-level communication parameters, such as sender, receiver and message identities. The specification of the FIPA agent communication contains message exchange interaction protocols, speech act theory-based communicative acts and content language representations (FIPA 2002b). The FIPA ACL message structure contains a set of one or more message parameters, depending on the situation (FIPA 2002a). The mandatory parameter is the *performative* which describes the type of communicative acts. Other parameters such as sender, receiver and message content are expected being contained in most ACL messages, too. The semantics of FIPA messages are defined in the Semantic Language (SL) (FIPA 2002b). Devices are described in ontologies using the FIPA Device Ontology Specification (FIPA 2002c). Based on the FIPA specifications, a number of agent platform implementations are available such as JACK® Intelligent Agents or JADE (FIPA 2003). For the overall organization of a MAS several organization patterns such as teams, coalitions, markets and also heterarchic and hierarchic (including holonic) architectures do exist (Monostori *et al.* 2006).

Architectures based on agent technology have been developed for various purposes of manufacturing. Engineering design for e.g. supporting the life-cycle management of products, process planning and production planning has been under investigation within several research projects and industrial applications.

The AARIA (Autonomous Agents for Rock Island Arsenal) architecture focuses on an agent-based system for shop-floor control and scheduling (van Dyke Parunak *et al.* 2001). AARIA defines three persistent agents (*Brokers*) for physical resources, parts and units. The authors derive from a set of requirements the design for an industrial-strength system, which has been developed for an Army manufacturing facility.

Within the MASCADA project (MASCADA 1997), a multi-agent system has been developed which is based on the PROSA architecture (see Section 2.3). It concentrates on the manufacturing execution system composed of communicating local intelligent agents. For the decision making process related to routing and processing, the agents distribute system state information (Valckenaers *et al.* 1999). Pro-active disturbance handling is used in order to react on unforeseen system component failures. MASCADA is particularly designed to manage production change and disturbance handling. The main test case of MASCADA was a section of a painting cell in the passenger-car plant of the Daimler-Benz AG in Sindelfingen, Germany. The applications consisted of two paining steps and recovery procedures in order to obtain a perfectly painted car (Brückner *et al.* 1998).

The ABAS (actor-based assembly systems) reference architecture defines new types of autonomous mechatronic units called *actors* (Lastra *et al.* 2009). ABAS uses auction- and negotiation-based multi-agent control in order to implement a reconfigurable manufacturing system (see also Koren *et al.* 1999). A highly-dynamic reconfigurable assembly solution including conveyors and transfer units based on ABAS has been demonstrated in a pilot installation in Tampere, Finland.

Terzic *et al.* (2009) propose a twofold approach for a multi-agent control system. A modular, knowledge-based architecture is combined with a diagnostic and user interaction infrastructure in which the user is modeled as an agent. The architecture of the infrastructure relies on the traditional three level approach, including ERP, MES and shop floor level. By keeping the hierarchy of traditional factory organization and at the same time by making the behavior and decisions of the agents more transparent, the authors assume to increase the acceptance of end users and operators for multi-agent systems.

The Production 2000+ (P2P) project of the DaimlerChrysler AG had the goal to enhance the flexibility, robustness and scalability of production. The agent-based control system FactoryBroker combines the advantages of MAS with those of the holonic manufacturing systems (see next section). FactoryBroker allows for the integration of very heterogeneous mechatronics and software components and is able to perform decision process in real time manufacturing environment (Colombo *et al.* 2005). A prototype of the agent-based approach has been installed in parallel to a conventional production line for the production of engines in a DaimlerChrysler facility (Bussmann&Schild 2001).

PABADIS'PROMISE system is based on the research activities of the PABADIS project in which a multi-agent system for the control level has been developed (Feng *et al.* 2007). PABADIS'PROMISE extended this approach by the introduction of a more agile system architecture (Fig. 6). In addition to the mobile agents on factory level, agents have also been introduced on the field level. Machines and devices are represented by resource agents who are able to communicate and negotiate on open manufacturing jobs.

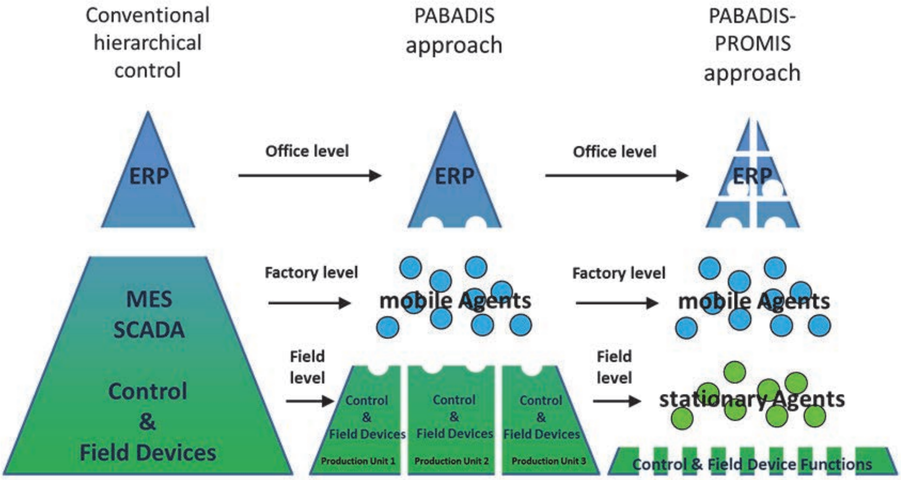


Fig. 6. Transition from conventional hierarchical control via PABADIS approach to PABADIS'PROMIS approach (Peschke *et al.* 2005, published by permission of © IEEE).

Order agents on MES level are located in RFID tags which are attached to the product to be produced (Peschke *et al.* 2005). In order to reach faster reaction time and closer control on shop-floor, PABADIS'PROMISE proposes to restrict the functionality of the ERP systems only for strategic planning and to move all tactical planning to the MES level. Furthermore, the monolithic architecture of the MES level shall be replaced by a set of independently acting, cooperative units (Lüder *et al.* 2006). For that, PABADIS'PROMISE proposes a completely decentralized architecture both on MES level and on field control level. The PABADIS'PROMISE approach has been tested at FIAT car body manufacturing for order option changes in real-time and for an assembly line reconfiguration (PABADIS-PROMISE 2008).

In the SIARAS project a skill-based approach to enable dynamic reconfiguration of complex production processes has been investigated (Eigenbrod 2008). It uses ontologies for the modeling of skills which are stored in a central *SkillServer*. The approach has been tested on a modular micro-production system (Bengel 2008). An application for the manufacturing and inspection of doorplates has been demonstrated at the Automatica Fair Munich 2008 (Idea 2013).

The SOCRADES project aimed on the development of a service-oriented framework for device-level infrastructures. The system intelligence is achieved by the composition of intelligent physical agents which are embedded in smart devices. (SOCRADES 2007a). To do so, semantic web-services are implemented on the devices. Agents can discover, select and composite devices for production orchestration. SOCRADES implemented several demonstrators, e.g. a wireless control and monitoring in ore concentration plant at Boliden premises or the integration of both real and virtual devices in a modular production cell at Schneider Electric Automation (SOCRADES 2007b).

At present, several projects under the 7th framework programme are carried out which are dealing with agent technology for manufacturing systems. The IDEAS project focuses on the implementation of agent-based distributed control on shop-floor (IDEAS 2013). IDEAS is based on the research activities of the EUPASS project and belongs to the category of evolvable assembly/production systems (Ribeiro *et al.* 2010, Onori *et al.* 2012). The self- and reconfiguration of production entities is done solely on shop-floor without requiring supervising components. A pre-demonstrator has been built in a test bed for the assembly of medical components (Ribeiro *et al.* 2011). The final demonstrators are built in industrial prototypes for the production of electronic components.

The GRACE project aims on the development, implementation and test of a collaborative MAS for process control and quality control at local and global levels (Castellini *et al.* 2011). GRACE is implemented on the MES level which is interacting with lower-level devices. The GRACE architecture is inspired by several established multi-agent systems architectures such as PROSA, ADACOR and PABADIS-PROMISE. The GRACE architecture shall be tested for various use-cases such as the geometry inspection of the washing machine drum or the self-creation and self-adaption of test plans in automatic quality control systems. (GRACE 2013).

The Skill-Pro project extents the *plug-and-produce* paradigm by discovering the skills of the production entities for their composition and cooperation. To do so, an *Asset Management System* is introduced on MES level which supports process planning, process control and process monitoring. Skill-Pro intends to evaluate its approach in real-world scenarios. (Skill-Pro 2013).

PRIME aims on multi-agent control, dynamic knowledge sharing and innovative human-machine interaction mechanisms in order to allow for highly adaptive, reconfigurable self-aware plug and produce assembly systems. The PRIME approach is supposed to integrate the process units from different suppliers in a plug and produce assembly system without forcing the competitors to reveal their process and technology know-how. PRIME intends to implement three industrial demonstrators in key assembly sectors. (PRIME 2013).

2.3 Holonic control architectures

Holonic manufacturing systems (HMS) have been seen as a new paradigm for intelligent, flexible manufacturing organization. It has been developed in the framework of the Intelligent Manufacturing Systems (IMS) initiative (IMS 2012). The word *holon* has been proposed by Arthur Koestler who developed concepts for social organizations and living organisms (Koestler 1989). Holon as a word is a combination of the Greek word *holos* meaning “whole” and the suffix -on which is its neuter form. Holonic control is supposed to combine the advantages of both hierarchical and heterarchical control while avoiding their drawbacks (van Brussel *et al.* 1999). Within the IMS initiative, the following definitions related to HMS have been carried out in order to translate Koestlers holonic concepts into the manufacturing environment (Christensen 1994):

- *holon*: An autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information and physical objects. The holon consists of an information processing part and often a physical processing part. A holon can form part of another holon;
- *autonomy*: The capability of an entity to create and control the execution of its own plans and/or strategies;
- *cooperation*: A process whereby a set of entities develop mutually acceptable plans and execute them;
- *holarchy*: A system of holons which can cooperate to achieve a goal or objective. The holarchy defines the basic rules for cooperation of the holons and thereby limits their autonomy;
- *holonic manufacturing system*: A holarchy which integrates the entire range of manufacturing activities from order booking through design, production and marketing to realize the agile manufacturing enterprise.

In the control architecture each holon is a self-controlling and self-executing entity which is cooperating by communicating and negotiating with all other entities in the system. Each holon can be composed of other holons which are building a holarchy. Hence, also self-similar elements similar to the fractal manufacturing systems (FMS) are present (Warnecke 1996). Holonic control architectures were seen as the consequent architectural evolution in manufacturing systems of the approaches illustrated in Fig. 4 on page 23. The benefits of such an evolution were seen in the fulfillment of the key architectural requirements of disturbance handling, availability and robustness as well as for human integration and flexibility (Christensen 1994).

Based on the quite generic description of holonic systems, several designs and reference architectures have been developed. The latter are supposed to provide coherent engineering and design principles for a specific domain, including a unified terminology, system's structure, system components and their responsibilities, etc. (Wyns *et al.* 1996). A major task in the design and the implementation of holonic manufacturing systems is the identification of the required holons. To assist the system's designers in the identification and development of holons, the ANEMONA methodology has been developed (Botti&Giret 2008). Based on a mixed top-down and bottom-up approach, ANEMONA defines an analysis phase including system requirements analysis, holon identification and holon specification as well as the holon design phase for the derivation and implementation of the specific holons based on the earlier analysis phase.

The division for Production engineering, Machine design and Automation at the K.U. Leuven (Belgium) developed a reference architecture called Product-Resource-Order-Staff Architecture (PROSA) (van Brussel *et al.* 1998). The PROSA architecture defines order holons, product holons and resource holons as its basic components (Fig. 7).

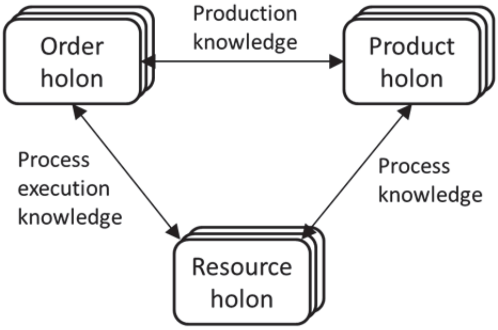


Fig. 7. PROSA building blocks and their relations (van Brussel *et al.* 1998, published by permission of © Elsevier).

Order holons represent the tasks in a manufacturing system and thus, perform the activities of dispatchers, progress monitors and short term schedulers of traditional manufacturing systems. Product holons keep the knowledge on the processes and the product for making the product in the desired quality. Product holons comprise the functionalities of the product design, process planning and quality assurance of traditional manufacturing. Resource holons contain the physical part of a manufacturing system. They are an abstraction of the production and thus, can be interpreted as energy, material, personnel, equipment, production lines or even whole factories. Staff holons can be added in order to add expert knowledge to the basic holons.

PROSA provides interesting aspects especially in self-similarity of manufacturing entities and in covering hierarchical and heterarchical manufacturing organization. The main innovations can be seen in decoupling the system structure from the control algorithm and logistical aspects from technical ones. PROSA focusses much on the control aspect of manufacturing systems and provides a generic reference architecture. A methodology for the implementation of a concrete manufacturing system is not part of PROSA. PROSA has served as a basis for several further architectures and system designs.

The ADACOR architecture comprises a model of four basic holons, namely supervisor holon, product, task and operational holons (Leitão&Restivo 2003). While the product, task and operation holons are quite similar to the basic holons of the PROSA architecture, the supervisor holons significantly differs to the PROSA's staff holon.

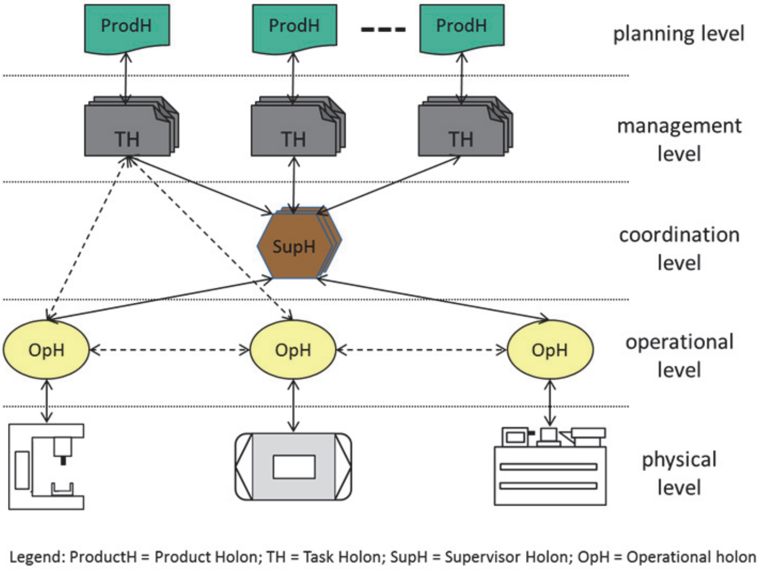


Fig. 8. ADACOR holon classes (Leitão&Restivo 2006, published by permission of © Elsevier).

The role of the supervisor holon is to coordinate the other entities in the system, to combine synergies, to aggregate the skills of each member of the group and to offer the combined services to other entities in the manufacturing system (Leitão&Restivo 2006). Supervisor holons can therefore not mapped onto or compared to the PROSA staff holons. In PROSA, a holon having similar functionality as the ADACOR supervisor holon would be an aggregated holon comprising supervisor functionality and operational holons as subholons. The coordination of the holons in ADACOR is done by ordinary Petri net models. The concept of the supervising holon allows for global optimization in decentralized structures and for an adaptive holonic control which enables for a dynamic self-reconfiguration of the control structure. Furthermore, ADACOR provides a concept for the integration of physical automation devices by using a virtual device concept. ADACOR has been tested in a production system layout which comprised physical manufacturing cells, an assembly cell, an inspection cell and a transport system extended with two virtual manufacturing cells for providing flexibility in achieving alternative production routings.

A holonic packing cell has been developed by the University of Cambrige using the JACK® agent platform (JACK 2013). The holons are supposed to represent the physical components of the system and Radio-Frequency Identification (RFID) technology is used to replace the traditional bar codes (Fletcher *et al.* 2003). The cell enables a customer to select three Gillette™ personal grooming products and also how they are to be packed into one of two box types.

Other developments of holonic architectures concentrate on the enrichment of the PROSA architecture by agent technologies. The CICA architecture uses component based software designs in order to obtain flexibility on system level. By this, advanced exception handling and reconfigurability can be reached. (Su 2007). The PROSAGENT framework extends the PROSA architecture by giving each Resource holon and Order holon an agent component. PROSAGENT proposes to build applications at two distinct levels, the holonic (object) level and the properly agent level. The object (holonic) level is supposed to control centralized production for mass production and the agent (distributed) level is switched on and assumes control of production processes when disturbances render central control impractical. (Hartonas 2006).

Another instantiation of PROSA has been developed inspired by natural systems. The basic design of PROSA has been enriched to support stigmergy. Stigmergy has been introduced by Grasse (1959) for describing the indirect communication mechanisms of social insects by signs which replace any direct communication between entities of a system. In nature, stigmergy can be found for example in ant colonies. To support such a behavior, information spaces are added to the PROSA resource holons which are under the control of the respective resource holon and which can be accessed by any other holon in the system. The major benefit of the system design is its capability of predicting the future behavior and of proactively taking measures in order to prevent impending problems (Valckenaers & van Brussel 2005). The implementation has been tested in industrial environment in a job shop producing weaving machine parts and in heat treatment factories.

The ManufAg framework includes distributed hierarchical decision-making schemes into the production control (Mönch&Stehli 2006). ManufAg has been tested in a case study called FABMAS. FABMAS is a hierarchically organized multi-agent-system for production control of semiconductor manufacturing processes (Mönch *et al.* 2006).

The multi-agent system NovaFlex is installed at the Intelligent Robotic Centre in UNINOVA. It is composed of two assembly robots, an automatic warehouse and a transport system that connects all the modules, multi-agent implementation to control a shop floor system. To do so, each shop floor component is agentified enhancing its adaptability and interaction competences to respond to environmental requests. Each agent representing a shop floor component can be aggregated to form a coalition that coordinates higher level processes (complex skills) based on the ones available in its members. In addition to that, an ontology is used to ensure an accurate information exchange as well as to define the domain and relations between entities. (Cândido&Barata 2007).

In Blanc *et al.* (2008) a PROSA-based manufacturing system for the production of security glass in a real industrial environment is presented. Security glass consists of layers of glass and plastic plates. The system consists of three holon types, namely Assembling holons, Disassembling holons, and Transforming holons. In the illustrated example, the Disassembly holon is supposed to allocation and placement problem of the glass layer into a cutting pattern and a scheduling problem of the cutting pattern of the plastic plate.

Within the IntelliFeed project, a holonic setup for a paint shop has been developed (Lind *et al.* 2009). The implementation of the setup concentrates on a part upload station. For that, robots and robot tools have been holonified. The authors mention their confidence in the HMS paradigm and recommend therefore the system for industrial applications.

2.4 Relevant industrial standards

Standards play an important role for the success of novel approaches in the industrial implementation (Leitão 2009). Even though the establishment of new standards is not intended – neither as part of the XPRESS project nor as part of this thesis – a short description of relevant industrial standards seems to be helpful for an implementation of the architecture proposed within this thesis.

The most important standards with respect to this thesis are the industrial control standards IEC 61131 and IEC 61499. The IEC 61131 standard was introduced in the early 1990s (IEC 2003a). Its aim is to define standards for the programming of programmable logic controllers (PLCs). IEC 61131 is currently divided into eight officially released parts concentrating on different standardization goals. For example, IEC 61131-3 describes the standardization of programming languages and IEC 61131-5 is about communication issues. The newest part IEC 61131-9 deals with interfaces for small sensors and actuators and is currently available as a pre-release of the official standard (IEC 2013).

In order to tackle the challenges of future industrial systems, IEC 61499-1 defines an open architecture for distributed and embedded control and automation (IEC 2005). Among others, IEC 61499 defines reusable modules (function blocks), event-driven execution and data. IEC 61499 was officially launched in 2005. Early case studies investigated the usability and performance of IEC 61499 (Gerber *et al.* 2008). By a wider adoption of IEC 61499, industry can also benefit from the new approaches in holonic and other sophisticated areas. On the other hand, barriers and doubts regarding a strong penetration do exist. For example, there are barriers with respect to determinism, performance, IP protection (due to open document standards) or lock-in effects (devices, software, development tools, etc.) and a great deal of time and effort would be required in order to obtain the same level of maturity as commercial PLC tools (Vyatkin 2011).

In addition to the control standards, the following standards should be taken into account for the industrial implementation of the architecture: IEC 61158 defines fieldbus technology for industrial network systems for real-time distributed control (IEC 2010a) and IEC 61784 defines a set of protocol specific communication profiles (IEC 2010b). IEC 61804 provides guidelines for device integration to meet requirements on compatibility, the interwork-ability, the interconnect-ability, the interoperability and the interchange-ability (IEC 2003b). IEC 61804-3 specifies the Electronic Device Description Language (EDDL) technology. EDDL is a generic language for describing the properties of automation system components in a syntax-independent manner (IEC 2010c). AutomationML (Automation Markup Language) provides an open standard for a neutral data format based on XML for the storage and exchange of plant engineering information (W3C 2006). This allows heterogeneous engineering tools and devices to be interconnected. Real components are described by objects which are able to encapsulate other sub-objects. As a result, compositions of very different detailing can be defined. The attributes of typical objects comprise the geometry, its kinematic, its behavior, its position within the hierarchical plant topology and the relations to other objects (Drath *et al.* 2008).

2.5 Summary and conclusions

In Chapter 2 the scientific state of the art of manufacturing control systems is described. In Section 2.1 the traditional control architectures are presented and their advantages and disadvantages are discussed. In Section 2.2 the agent-based concepts and architectures are illustrated. After the definition of agent-based manufacturing systems and their elements, various approaches for such systems are described in more detail. After that, the holonic manufacturing systems are presented in Section 2.3. A short description of the relevant industrial control standards completed the state of the art review.

The review illustrates the fundamental concepts for future manufacturing systems and provides an overview of the past and current efforts for brining those approaches into practice. In Table 1 on the following page the various approaches, their application or test environment and the level of implementation is summarized. It can be seen, that a number of test beds, laboratory and industrial prototypes as well as real industrial applications have been implemented on the basis of agent and/or holonic architectures for manufacturing systems.

Even though their improvements in terms of performance, flexibility and robustness have been successfully demonstrated, a wider adoption in the industry is still missing. The reasons for the unsatisfying uptake are manifold. Leitão (2009) identifies two groups of reasons for the slow adoption of intelligent approaches. The first group is related to the efficiency of the conceptual design of the approaches. Higher investment costs for decentralized approaches in comparison to centralized ones, the need of the industry for technically approved technologies and a widely spread general fear of taking up new technologies hamper the implementation.

Table 1. Existing approaches for manufacturing systems, their application environment and the level of implementation.

Project/ architecture	Short description	Application/test environment	Level of implementation
AARIA	Agent-based system for shop-floor control and scheduling	Army manufacturing facility	Industrial prototype
MASCADA	Multi-agent system especially designed for production change and disturbance handling	Painting cell in passenger-car plant of the Daimler-Benz AG in Sindelfingen	Industrial prototype
PABADIS' PROMISE	MAS including manufacturing ontology, agent platform, RFID's, field control devices and ERP tools	FIAT car body manufacturing for order option changes in real-time and assembly line reconfiguration	Industrial prototype
CICA	Component-based control architecture for system-level control of reconfigurable manufacturing systems	Flexible manufacturing systems lab at Virginia Tech. In	Laboratory prototype
PROS-AGENT	PROSA extension with holonic level and the agent level	n.a.	n.a
Production 2000+	Agent-based material flow control	Assembly of engines at DaimlerChrysler in Stuttgart	Industrial production
JACK®,	Holonic cell control based on the Jack® agent platform	Packing cell at the University of Cambridge	Laboratory test bed
FABMAS	Holonic system for manufacturing control	Control of semiconductor wafer fabrication at University of Ilmenau	Industrial prototype
NovaFlex	Agent-based shop floor control	Transport and assembly cell at Intelligent Robotic Centre in UNINOVA, Lisbon, Portugal	Laboratory test bed
ADACOR	MAS control system using supervisor holons for coordination	Transporting and handling cell at the Polytechnic Institute of Braganca	Laboratory test bed
ABAS	Reference architecture for reconfigurable systems based on new types of autonomous mechatronic units	Micro assembly solution including conveyors and transfer units in Tampere, Finland	Laboratory test bed
GRACE	Process and quality control in a collaborative Multi-Agent System	Washing Machine production line at Whirlpool Europe in Naples, Italy	Industrial prototype
IDEAS	Agent-based distributed control on shop-floor.	Medical product assembly Electronic component assembly	Test bed Industrial prototype
SOCRADES	Semantic web-services on smart devices on shop-floor	Several applications, e.g. Ore concentration at Boliden, Sweden; Electrical component production at Schneider Electric, France	Industrial prototype
SkillPro	Asset management Systems for the discovery of skills	n.a.	n.a.

Project/ architecture	Short description	Application/test environment	Level of implementation
PRIME	Multi-agent control, knowledge sharing monitoring and human-machine interaction for adaptive assembly systems	n.a.	n.a.
SIARAS	Skill-based approach for reconfigurable manufacturing	Micro-production system	Test bed
PROSA-based (1)	PROSA-based architecture with Assembling holons, Disassembling holons and Transforming holons	Laminated security glass production for automotive applications at American Glass Product	Industrial production
PROSA-based (2)	PROSA-based architecture with special implementations of Product, resource, order and staff holons	Process and material handling in a paint shop	Laboratory prototype
PROSA-based (3)	PROSA-based architecture enriched by stigmergy elements in order to reach predictive and proactive control of production	Weaving machine parts production and batch building optimization of heat treatment factories across factory boundaries	Industrial prototype

Furthermore, decentralized systems require new approaches and ways of thinking at both, development and application side. With respect to this, end users hesitate to shift responsibility of component development and maintenance to the respective component suppliers. In addition to that, missing standards, proprietary tools and software applications in a heterogeneous environment are causing problems. Finally, most of the research efforts concentrate on pure software solutions without interfacing physical devices. In reality, only with the integration of physical devices in the respective architectures the entire manufacturing system can be adopted to the new approaches.

The second group targets to the development related aspects. For the design and implementation of complex systems, proper tools are required in order to understand their behavior. Additionally, current industrial controllers do not support sophisticated approaches, e.g. multi-agent implementation. Furthermore, limitation in the scalability of current laboratory prototypes e.g. in agent instantiation are reported. The principle design pattern of entities which allow for self-similarity such as holons is not fully exploited in current developments and implementations. Finally, an approval of capabilities in disturbance handling of agent-based and holonic systems is not extensively done so far.

Su (2007) mentions the complexity of the holarchies and the unpredictable behavior of holon cooperation as one main issue when implementing holonic systems in real factory environments. Furthermore, the top down approach of most systems leads to the need for complete re-organization of the factory management and a step-by-step implementation is mostly not possible. The contradictoriness of the holonic system implementation and the heterogeneous environment of automation and information systems of real factories has also been mentioned by Su (2007) and by Babiceanu (2005).

The lock-in effects mentioned in Vyatkin (2011) which hamper the smooth transition of IEC 61131 to IEC 61499 could also play a significant role in the introduction of agent-based or holonic manufacturing systems in the industry. In the same context Gerber *et al.* (2008) expresses the need of coexistent classical and advanced technologies as well as the stepwise approach for the introduction of new technology.

Sundermeyer&Bussmann (2001) list in their report on the experience on the introduction of agent technology in a productive environment several critical success factors for a successful implementation: A consequent cooperation of end users, system and components suppliers and research is required in order to tackle the technological challenges. Furthermore, potential for improvements, technological feasibility and industrial readiness need to be demonstrated in each phase of the integration process in order to minimize the risks on investments and to gain acceptance at the end users side. The final decision on the implementation of new technology relies on the outcome of a cost-benefit analysis.

2.6 Major research questions and research methodology

As illustrated above, there are several aspects which hamper the successful implementation of holonic and/or multi-agent systems in the industry. This thesis aims to contribute to a wider adoption of such approaches by proposing a novel architecture for flexible manufacturing systems. Even though the system architecture also covers the planning (ERP) and control level (MES) of manufacturing systems, the main goal is to overcome the weak contribution of existing research approaches on device integration at shop floor level. Against this background, this thesis concentrates on answering the following major research questions:

- What are the standards, restrictions and boundary conditions required by the industries under investigation in terms of flexible manufacturing systems and how do they relate to the strategic roadmaps for future manufacturing systems?

- How could flexible devices look like which also take into account the restrictions elaborated by the industry?
- What are the characteristics of a manufacturing system architecture that allows for the integration of devices and which meets the requirements?
- What are the strengths and weaknesses of the architecture? What is the price to be paid when implementing such a system architecture?
- Does the developed architecture really work in practice? How much effort is required to make it work? Where are the limitations of the architecture in general and within the several validation scenarios?

The underlying research methodology which leads to the answers to these research questions is applied as follows: Firstly, the industrial requirements in terms of future flexible manufacturing systems are analyzed, consolidated and generalized. Major input is taken from the requirement gathering process of the three industry types involved in the XPRESS project. During this process, information from various workshops, interviews and other methods were used by the author who was also responsible for the subsequent consolidation and generalization of the requirements. A further input source is a literature search for the characterization of the processes and production flows in the sheet metal industry, in additive manufacturing and in the heat treatment industry. In the second step of the methodology, an architecture for flexible manufacturing systems is delimitated. This is done by taking advantage of existing scientific approaches and by combining and enriching them to an architecture which meets the industrial requirements. The major building blocks of the architecture are created in a third step. As mentioned before, the focus herein is on research on the integration of intelligent devices at shop-floor level. The architecture as well as the sample implementations of intelligent devices related to welding processes used to test the feasibility of the approaches are penned by the author of this thesis. The final step in the research methodology is the validation of the architecture using several industrial demonstrators implemented by and at the facilities of the respective partners of the XPRESS project. The efforts of the adaption of the architecture to the different demonstrators are assessed against the revealed benefits. In addition, the architecture is validated against the requirements of the sheet metal industry, additive manufacturing and the heat treatment industry.

3 Requirements for the manufacturing system architecture

This chapter describes the requirements for future manufacturing systems from the view point of representatives of three industry branches. The first section of this chapter provides an overview of the requirement gathering process. It illustrates how the process has been carried out, which industries have been involved and how the requirements have been analyzed. After that, the output of the requirements gathering process is discussed and a consolidation of the requirements is done. Based on that outcome, the main features of future manufacturing systems for traditional-oriented industry branches under consideration in this work are derived. In the next section, a short discussion of the requirements and their relation to common demands and strategic roadmaps is done. The last section of this chapter provides a characterization of additional three industries based on a literature research. The proposed architecture will later on validated with respect to their processes and process flows.

3.1 Requirement gathering process

The requirement gathering process has been carried out during the starting phase of the European Research Project XPRESS. The main goal of the requirements gathering was to identify the most urgent needs of the industry for the capabilities of future flexible production systems. For that, the gathering process was strongly industry-driven. Representatives of three different industry branches have been involved:

1. Aerospace industry with representatives of a manufacturer of airplane fuselages (end-user), of a component supplier and system integrator of riveting machines as well as a research institute specialized in riveting process optimization working closely with the industry.
2. Automotive industry with representatives of a car manufacturer, a robot supplier and system integrator specialized in the design of entire production lines, a system integrator specialized in the design and implementation of handling and joining production cells, a component supplier for welding control units as well as a company specialized in welding simulation and optimization software.

3. Electrical industry with representatives of a company specialized in designing and assembling of machines for the manufacturing of electronic components such as switches and relays, a component supplier for welding equipment as well as a company specialized in the programming of process and production flow software.

Table 2 illustrates the very different properties of those industries by providing an overview of the main characteristics.

Table 2. Characterization of the aerospace, automotive and electrical industry.

Criteria	Aerospace	Automotive	Electrical
Major processes	Transportation, clamping and riveting in body shop; painting and assembly afterwards	Transportation, handling, fixing, gluing in body shop; painting and assembly afterwards	Transportation, welding, brazing, assembly; Often additional processes such as molding
Process flow organization	Project or Job-Shop	Flow shop with some job shop elements	Pure flow-shop
Lot sizes	1 (almost each aircraft is unique)	1 (customization after body shop)	>1.000 (batch customization)
Cycle times	Several days	1 day	Minutes
Production strategy	Customer specific	Build-to-order	Batch production
Level of automation	High, with partially manual operations in tool handling and quality monitoring and assessment	High automation level in body shop, robot-assisted manual operations in final assembly	Very high, manual operations usually only for loading and unloading of the products
ICT usage	PLC-based solutions for processing and operator assistance on the single stations	Complex, networked systems; real-time busses and devices, databases, monitoring systems	Solutions based on one central PLC's. Individual cabling and programming for each machine.
Model change	>10 years	Between 5 and10 years	<2 years
Product characterization	Very large and heavy products; Up to 10m x 7m	Medium to large products	Small products

The requirement gathering process itself has been carried out in different steps (Hoffmeister 2008). In order to have a solid basis of the subsequent analysis, a large number of workshops have been performed together with the industrial partners. The workshops themes were related to the respective industry branches. In other words, "aerospace industry", "automotive industry" and "electrical industry" workshops have been performed. As the components suppliers deliver their products often to more than one industry, the representatives participated in various workshops of different industrial branches. By this, differences and commonalities among the industries could be identified in a very early stage. Within those workshops so-called *scenarios* have been gathered which are supposed to describe typical challenging situations of production. Mind mapping methods turned out to be the most efficient way for this. After the workshops the scenarios have been sorted and aggregated in order to identify similar and overlapping challenges across the three industry branches. Next, several use-cases and requirements within the scenarios have been worked out. For that, requirement documentation templates have been used in order to standardize the description and documentation of each requirement (Peschl 2008b). Finally each standardized requirement has been assigned to each scenario.

In addition to that an impact assessment for each scenario has been done in order to identify the impact of the implementation and improvement of the scenario on the six categories a.) Factory planning, b.) Product planning, c.) Pre-production and Laboratory phase, d.) Production execution e.) Sharing expertise among the different stages and f.) Quality monitoring, assessment and documentation (Hoffmeister 2008). Within this context, the thesis' author was responsible for accompanying the requirement gathering process at the different industrial partners and was also responsible for the alignment of the consolidated overall requirements with the project scope and vice versa. As an outcome of the consolidation process, four major demands have been identified. Those four major demands build the basis of the architecture for a flexible manufacturing system and are subsequently described in the next sections.

3.2 Flexibility in production

The first demand is the need for having flexible production systems. All representatives involved in the requirement gathering process expressed the urgent need for systems providing flexible behavior. However, the current situation and the targets of flexibility across the industry branches are very different. Within the next sub-sections, the various targets of flexibility under investigation in the requirements gathering process and their relation to the respective industries are illustrated.

[1.1] Flexibility of the products to be produced

Future manufacturing systems shall be able to produce multiple variants of one product within one production line, production cell or a machine respectively (Fig. 9).

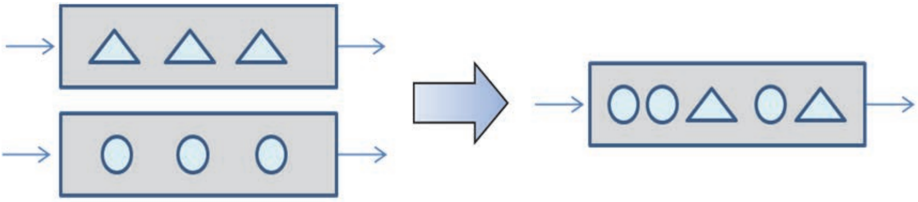


Fig. 9. Product flexibility: One production line for each variant (left) and one line for both variants (right).

Such manufacturing systems must be enabled for rapid change of production setup in order to minimize the downtime for changing over the equipment. By this, very small lot sizes down to lot size 1 shall be possible while also keeping the costs to an acceptable limit.

Having such systems, new markets can be targeted by product differentiation and the tailored needs of customers can be satisfied by mass customized products. In addition, to effect on the capabilities of the control systems and equipment itself, also the data handling, especially the data required for quality assurance of the product produced, has to be taken into account. With respect to the three industry branches, the special demands are different. In automotive industry, multi-variant production is already state of the art. However, significant efforts are required in order to reconfigure the production line for new products. The representatives of the automotive industry therefore have the fast reconfiguration of existing production lines and the rapid introduction of new equipment into exiting lines in focus. In aerospace industry almost each product is unique. Different customer's demand, e.g. in the cabin interior do also affect the setup of the processes for the production of the fuselage. For that, handling, transport, gripping and joining processes have to be adapted. Nowadays, these adaptations are often done manually by user interaction. The goal of the representatives is the replace the manual operations by an automatic adaptation on the new demands. In contrast to this, the machines used in electrical industry are designed for mass production. The production of a product mix on one machine is usually not possible and the changeover of a machine from one product to another is – if at all

– only possible with high efforts. The central goal of the electrical industry representatives therefore was to enable multi-variant production on the same machine by also keeping the costs reasonable.

[1.2] Routing flexibility

The routes of the products within the production process shall be flexible as shown in Fig. 10. Nowadays, the manufacturing systems of the industry branches involved provide only fixed production sequences. However, in case of machine breakdown or full capacity, the entire production process is blocked.

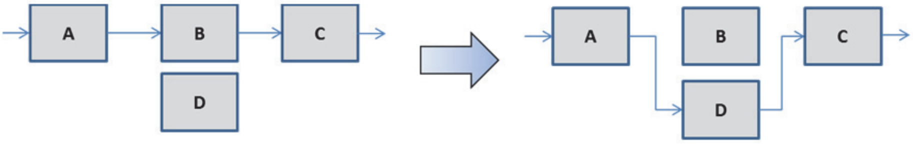


Fig. 10. Production flow flexibility: Standard production flow (left) and alternative route (right).

In such cases, the production flow shall be rerouted and alternative production equipment shall be used. Furthermore, in case of an identified product quality problem, rework shall be possible in such way that, e.g. a production step is repeated automatically. In case of very insufficient product quality the related (semi-finished) product shall be sorted out. In such cases, the regular production flow must be left and alternative routes must be found. Within the XPRESS consortium it turned out that such features are mainly asked in (semi-) automated environments like aerospace and partially automotive. In such cases, where workers either are responsible for the transportation of the semi-finished products or when workers perform manual operations on the products to be produced and thus, replacing automatic processes, the routing flexibility is seen as most important. One could also imagine using routing flexibility in combination with Automated Guided Vehicles (AGV's). But in XPRESS no industry partner took the responsibility for driving this. Instead, a University which is also specialized in unmanned submarine research and development took over this part.

[1.3] Flexibility of the production resources

The analysis of the requirements illuminated a big consensus with respect to the flexibility of the production resources. Every industry partner pointed out that the assignment of machines, single devices or sensors as well as human personnel located on shop-floor is at present days too static. Instead of this new production resources shall have the capability for:

1. Easy integration, e.g. in order to increase the capacity of the production in case of an identified bottleneck (expansion flexibility).
2. Easy exchange with the same resource type, e.g. in case of device breakdowns where the defect device is replaced by another device of the same type (machine flexibility).
3. Easy exchange with another resource type, e.g. when a defect robot for handling tasks is temporarily replaced a human worker.

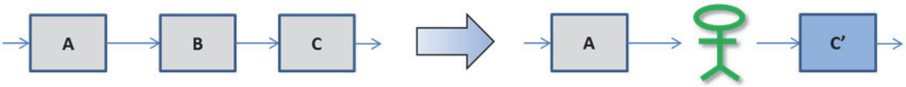


Fig. 11. Production resource flexibility: Usage of standard production resources (left) and of alternative resources (right).

In case of machines, devices and sensors, the features mentioned above are commonly known as *plug & produce* behavior (see also Hildebrand 2005). The interfaces and setup as well as the geometry of such equipment must be standardized in order to guarantee smooth integration or exchange without costly time-intensive ramp-up. If humans are supposed to overtake a role of a device, also other factors such as social or environmental aspects have to be taken into account. However, the human integration shall also be seamless without much reconfiguration efforts at the workplace (Fig. 11).

[1.4] Machine flexibility

The setup of devices at present days is usually done once during the ramp-up of production. During this phase, the device setup is optimized according to the actual condition of production. However, during the production lifetime, the conditions can change. For example, disturbances on the processes due to mechanical wear can lead to quality problems at the products.

For that reason, the setup of the devices shall be flexible in order to react on process disturbances and other changing boundary conditions. If for example the welding process is influenced negatively due to cooling problems, the devices shall have the capability to react flexible on such disturbances by an automatic (real-time) adaption of their process parameters (Fig. 12).

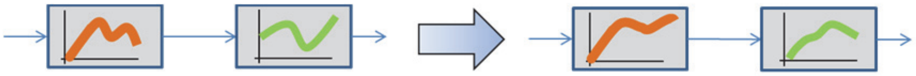


Fig. 12. Device setup flexibility: Standard setup of devices (left) and adapted setup in case of disturbances and changed boundary conditions (right).

The automotive system integrators who build production lines for different car manufacturers mentioned the very different philosophies of their customers: While some customers allow self-adaptive process setups without almost any limitations on the variation of the parameters, others rely on static process setup only without a minimum of variation. Thus, devices and processes are required which allow for a restriction of the degree of freedom for the self-adaption in order to guarantee that the process parameters do not exceed a requested limit.

[1.5] Flexibility of the production optimization goal

Nowadays, the setup of the entire production flow is done often once at the start of the production. The setup is mostly done to fulfill a specific optimization target, e.g. to have a good balance between the production output rate and the durability of the equipment (Fig. 13 left). In automotive body work, the optimization of the cycle time on the one side and the quality of the joints (which usually comes along with longer process times) are competing.

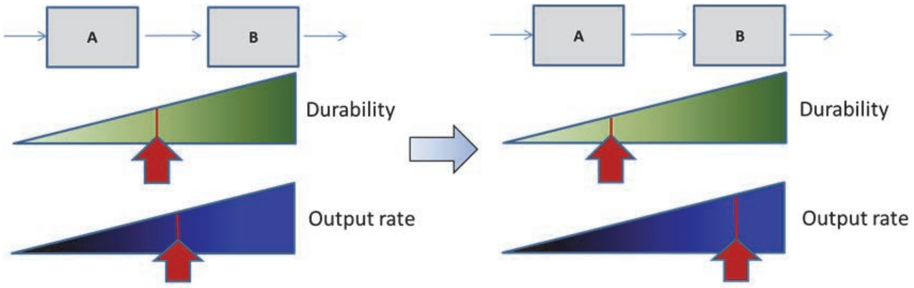


Fig. 13. Optimization goal flexibility: Medium durability of machines and medium output rate for normal production volumes (left) and low durability and high output rate for high production volumes (right).

However, the optimization goal can change. For example, when high production volumes are demanded, it is required to change the optimization target towards a high output rate which negatively influences the durability of the equipment (Fig. 13 right). Furthermore, other optimization targets can come into play in future. Especially in those industries which consume much energy in production, the optimization of the manufacturing system in terms of energy consumption could be useful. For that reason, future production systems shall have to have the capability to react flexibly on different optimization targets.

3.3 Handling expertise and knowhow

Nowadays manufacturing systems consist of complex and complicated organization and operational structures. For the design, construction, ramp-up and operation of such systems, many experts with very different technological background and expertise are required. During ramp-up and operation, various software systems, databases, machines and devices must work smoothly together in order to guarantee an optimal production process. By the introduction of more flexibility in the entire manufacturing process, the level of complexity is even more increased. If e.g. a new product variant (which was not foreseen when the respective production facility has been planned) shall be introduced in the manufacturing process, various adaptations of the related equipment, control systems, monitoring systems, workers and machine operators are required. A key factor in handling complex processes is the expert knowledge which is required. The next sections describe the requirements for handling expertise and knowhow from different points of view.

[2.1] Conservation and interlinking of knowhow and expertise

Knowhow and expertise is nowadays strongly related to the human who is an expert in a specific process or field of operation. If an employee leaves a company, his/her specific knowledge is also gone. This is in particular a problem if only a single employee or a small group of employees within a company is/are expert(s) in a specific knowledge domain.

The respective expertise is very distributed as it is spread among different humans. Huge benefits can be expected, if relevant expertise is available and accessible at every time and from every place. For the reasons mentioned above, it is required to conserve and to interlink relevant expertise in such a way that it is available for future usage. In addition to that especially the component suppliers involved in the requirement gathering process mentioned the high relevance of exploiting their specific knowledge on the processes performed by their devices and machines. Nowadays the added-value gained is based mainly on selling devices and in providing maintenance contracts. However, a huge benefit is seen in more service-oriented business models in which the main added-value is gained by exploiting knowhow on process-specific tasks such as rapid parameter finding, process optimization, quality assurance, diagnosis and documentation of the respective processes.

[2.2] Consolidation of distributed demands

Different stages are required before a product can finally be produced. At the beginning is the developments/design of the product and respective production system in which the product will be produced. In the ramp-up phase, the manufacturing system is installed and optimized. After that, the full capacity of the manufacturing system is reached and is ready for production execution.

For optimal production of a product, the demands of the experts involved in all stages of production must be taken into account. However, the demands and respective expertise are very distributed in terms of time (design, ramp-up, execution of the production process) and location (designers, operators, maintenance staff, etc.). Fig. 14 illustrates this situation.

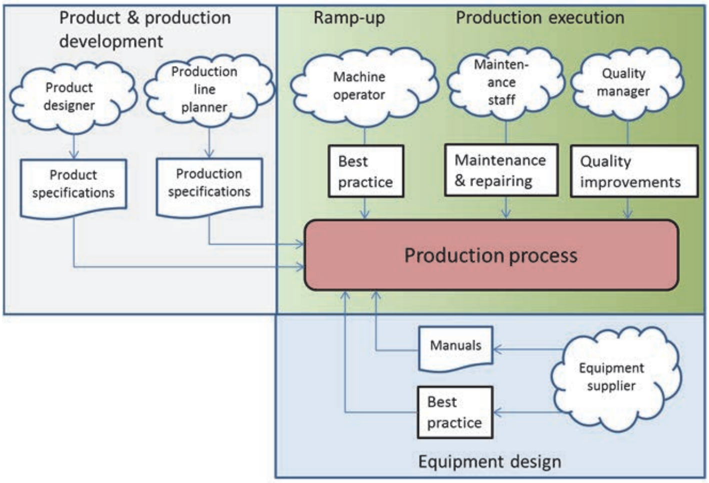


Fig. 14. Demands in different phases of production and distributed knowhow.

[2.3] Recipe-based solutions for knowledge conservation

Nowadays often solutions based on recipes are used for the configuration of the production. Recipes contain the minimum set of information about the development of a specific product (Oracle 2011). Production environments which have to deal with a lot of different products and product variants therefore require an intelligent and highly sophisticated recipe management. The outcome of the requirement gathering process has clearly shown that especially the smaller industries struggle with such solutions as the investment costs and efforts for maintaining such systems are too high. As a consequence, more easy and reliable solutions for product variant handling are required. The static recipe-based handling of the setup of production for the product variants has to be replaced by a solution which allows for variant handling in production without high efforts and costs.

3.4 Quality assurance

The requirement gathering process indicated that future manufacturing systems have to face the challenge of very sophisticated quality control, assessment and documentation. Even in nowadays manufacturing sites, increased efforts on quality assurance is required, e.g. due to new legal aspects. However, for future flexible manufacturing systems, the efforts on quality monitoring and assessment need to be increased even more: The more dynamic the involved processes will be performed, the more potential risks of failures can be expected. The next section describes the requirements on quality assessment and control for future manufacturing systems.

[3.1] Quality monitoring and assessment of all production processes

Nowadays often manual methods for testing the quality of the processes on shop-floor are used. In future, these shall be replaced by automatic assessment methods and systems. The assessed quality values and the descriptive data such as timestamp, equipment data, etc. shall be stored in a database for future use.

[3.2] Comparability of quality values

Nowadays quality assurance methods often generate quality values which are hardly comparable, even for the same processes. In resistance spot welding for example, manual destructive quality inspection methods deliver the spot diameter or the shear strength of the welding spot. Besides this, various proprietary quality values coming from the different equipment suppliers for welding processes are used, e.g. “Q value” (HWH 2009) or “Nugget index” (Matuschek 2012). However, an easy and automatic way for comparing those values does not exist. Future quality monitoring and assessment systems shall therefore have to have the capability to deliver standardized, comparable quality values.

[3.3] Relation of process and product quality

The quality requirements on a product such as geometrical tolerances or static and dynamic stiffness are usually specified during the product design phase. However, the reason for insufficient quality of the product cannot be found because the quality monitoring of the different productions steps such as joining or handling is related to the respective processes but not to the product. In the resistance welding process for example, a common quality assurance method is to monitor the welding signals such as voltage and current. If those signals run out of a predefined range, an insufficient welding quality is assumed. However, whether this process behavior really affects the quality of the product cannot be assessed.

Future manufacturing systems shall have the capability to relate the process performance and quality to the quality of the respective product.

[3.4] Product quality tracking

Nowadays production often also lacks of reliable product tracking. There are often no mechanisms to track and to identify the single product or product components within the production process. During production, the related quality measurements are usually done randomly at the end of the related production step. However, by the local assessment of quality at the actual production steps, no relation to the overall product quality is possible. For that it is required to track the product during the entire production process and to assign the quality of each single production step to the product.

[3.5] Product and equipment optimization

Nowadays, the design of a product, the design and construction of the production system and the setup of the single processes are done independently from each other. The reasons for this have already been described in Section 3.3. However, major benefits are expected if those steps are interlinked. Starting from the assessment of the single processes, the overall production system can be optimized, e.g. in terms of cycle time or overall quality. Furthermore, process feedback can also be used for the optimization of the design of a product. On the basis of process quality data, for example, weaknesses in the structure or geometry of the product can be identified which allows for subsequent optimization of the product design. Finally, feedback of relevant process and quality data also to the component suppliers enables them to optimize their equipment. Relevant data for such an optimization is information on the result of process execution or on maintenance activities on the equipment. By this it is possible to e.g. identify (gradually) wear of equipment. Fig. 15 illustrates the different optimization loops based on the assessment of process quality data.

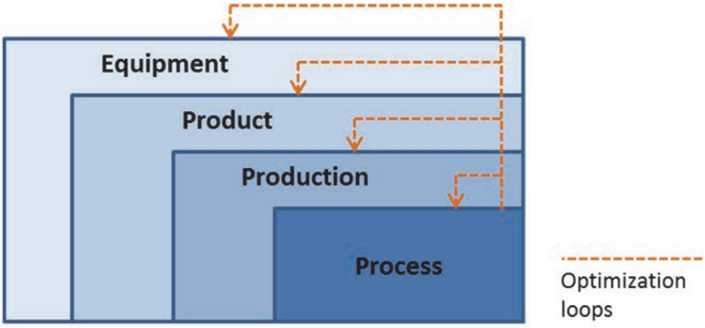


Fig. 15. Potential optimizations based on the assessment of process quality data coming from the shop floor.

Via data mining for example it is also possible to find hidden interrelations in the data coming from different equipment. The basis of the optimization opportunities described above is to gather and to assess the data measured during the process execution.

3.5 Factory organization

The last domain under consideration during the requirement gathering process targets the investigation of the future needs of the general organization of the factory. The main aspect in this domain is to answer the question, how future manufacturing systems shall be organized by taking the special requirements of traditional-oriented industries into account.

[4.1] Manufacturing organization

Especially the experts from the automotive and aerospace industry pointed out, that their way of manufacturing has been used and optimized during several decades. They are relying on the traditional organization of the automation pyramid which is divided into ERP, MES and shop-floor level (see e.g. Wollert 2006). They pointed out that they are not willing to accept new technology which does not fit into the hierarchical system organization. Thus, future manufacturing systems must be able to fit into the traditional three-layer structure of factory organization (see Fig. 16).

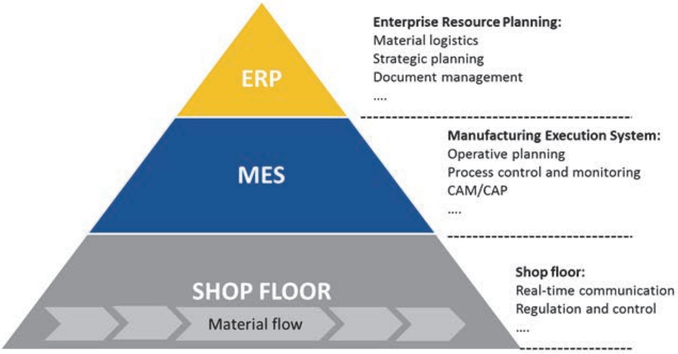


Fig. 16. Traditional automation pyramid with the levels ERP, MES and shop-floor.

Furthermore, many people hesitate to introduce completely new and radical solutions in one step due to the high risks related to this. Therefore it is also required to introduce novel technology in a step-by-step approach. Each step shall have a significant positive impact in order to convince people on the new approach.

[4.2] Using and embedding existing equipment on shop-floor

For carrying out the different manufacturing processes such as handling, joining or transporting, a huge amount of hardware and related software modules are required. Often, also customer-tailored versions of such equipment are used in order to fulfill the special requirements of the respective manufacturing process. Due to the high investment costs it is required that future manufacturing systems are able to embed already existing equipment in a smooth and seamless manner. Vice versa, equipment suppliers must be enabled to adapt their products easily in order to meet the requirements of novel manufacturing organization.

[4.3] Segregation of responsibilities in shop-floor organization

In the building and the ramp-up phase of production, suppliers are often responsible for a certain production line/cell/equipment. A strong hierarchical order is established. Fig. 17 provides an example from the automotive industry.

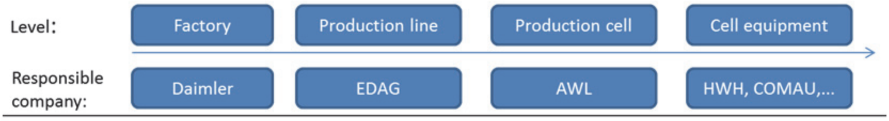


Fig. 17. Example of responsibilities for different levels, domains and components in the automotive industry.

During operation workers and operators have clear responsibilities which are restricted to production lines, cells or even single machines. Workers and operators target on a best operation within "their" environment. The operation of other domains is not (that much) in focus. In order to guarantee a smooth take-up of a novel approach in manufacturing, the separation of responsibilities has to be taken into account. Workers and operators must be convinced that a new approach respects the traditional way of responsibility segregation. Otherwise, the acceptance for the introduction of a new system might be low.

[4.4] Deterministic behavior of process execution

Even if very flexible manufacturing systems are required, the requirement gathering process indicated that there are some restrictions in flexibility which has to be taken into account. Due to just in time production, a deterministic behavior of the manufacturing process and relatively fixed production cycles which guarantee a minimum output rate are required.

[4.5] Reproducibility

In addition to the deterministic behavior, also a high reproducibility of process steps and process execution is sometimes required. The requirement gathering process turned out, that for example, some manufacturers allow very flexible process behavior in that sense, that controls and devices are allowed to adapt their settings during process execution in wide ranges while other companies insist on fixed control settings. The degree of freedom is therefore often a question of the company's philosophy.

3.6 Consolidation and conclusions on the requirements

In this section, the topics identified within the previous four sections are consolidated. The consolidation of the discussion topics identifies ten fundamental requirements for a manufacturing system for the applications under investigation. For each of them, the reference to the respective topic(s) mentioned above and the main drivers are given. At the end of this section, the conclusions on the requirements are provided.

REQ 1. Flexibility: Fast and easy introduction of new products

The first requirement is related to the ramp-up of production. The novel system shall allow for fast ramp-up for new products and product variants. Mainly involved are the re-organization of the production flow on MES level and the setup of the machines and devices on the shop-floor. With respect to the required limitation on dynamic system behavior and REQ 6, system and device re-organization shall be done rather within planning scenarios than in dynamic self-adaptation of system components.

Reference to: [1.1] Product flexibility; [1.2] Routing flexibility.

Main drivers: Automotive end users and automotive system integrators.

REQ 2. Reconfigurability: Reaction on changing demands in production

The manufacturing system shall be able to react fast on changing demands, e.g. production volumes, product variants or desired quality goals. In contrast to REQ 1, here only minor changes in the system are required. This mainly targets on the capabilities of the machines and devices on shop-floor on self-adaptation. In addition to that, the integration of humans on shop-floor must be possible without high reconfiguration efforts at the workplace.

Reference to: [1.1] Product flexibility; [1.2] Routing flexibility; [1.3] Resource flexibility; [1.4] Machine flexibility; [1.5] Optimization goal flexibility; [2.3] Recipe management.

Main drivers: End users of aerospace, automotive and electrical industry.

REQ 3. Learning ability: The manufacturing system and its components shall be enabled for continuous improvement

All components in the manufacturing system shall have the capability for continuous improvement and learning. This targets a frequent optimization of the manufacturing system and its devices. For that, sharing of best-practices between the various system components is also needed (see also REQ 8).

Reference to: [2.1] Conservation and interlinking of knowhow and expertise; [3.5] Product and equipment optimization.

Main drivers: End users of aerospace and automotive industry as well as the respective component suppliers.

REQ 4. Scalability and extensibility: The manufacturing system's control architecture shall be useful in different industrial environments and shall be able to adapt to changing demands

The industries involved in the requirement gathering process illustrated above do have very different system setups. Large manufacturing systems are in use in aerospace and automotive industry. In electrical industry small setups of manufacturing systems within single machines are used. The principals of the novel manufacturing system shall therefore be usable in different industrial sectors and should be scalable to different setups. Furthermore, it shall be extensible. If e.g. a machine for the production of electrical components based on the novel manufacturing system is integrated into a larger environment, it shall be easy to merge both setups into one environment (by also respecting the boundaries; REQ 7).

Reference to: [4.1] Manufacturing organization.

Main drivers: System integrators of aerospace and automotive industry.

REQ 5. Re-use of existing systems: Existing systems shall be (re-)used and a step-by-step implementation shall be possible

In order to improve the acceptance of a novel system and to limit the monetary risks, the manufacturing system shall allow a step-by-step implementation. It shall be possible to re-use existing implementations of manufacturing systems and to introduce the novel approaches step-by-step. Furthermore, it shall be possible to update or to interface existing equipment and software systems in order to minimize the investment costs.

Reference to: [4.1] Manufacturing organization; [4.2] Using and embedding existing equipment on shop-floor.

Main drivers: End users of aerospace, automotive and electrical industry.

REQ 6. Deterministic behavior: The manufacturing system and its components shall react predictively on changing boundary conditions

Even if the manufacturing systems control architecture shall allow the introduction of new products and shall also allow for reaction on changing demands, the reaction of the system needs to be deterministic. In particular this means that production cycles times can be guaranteed within a fixed interval and that the desired production output rate can be guaranteed. Against this background, different tendencies have been investigated: While the automotive industry relies on relatively fixed cycle times (due to strict just-in-time production and logistics), aerospace and electrical industries are willing to accept fluctuations in the processing times of a single product provided that the distribution of the expected variation can be predicted. As a consequence, the system's dynamic behavior needs to be restricted and a prediction of the potential variations must be possible. Slight changes can be probably accepted while major modifications on the system might be simulated in order to prove the system's behavior on the changes in advance.

Reference to: [1.1] Product flexibility, [1.2] Routing flexibility; [1.4] Machine flexibility; [1.5] Optimization goal flexibility; [4.4] Deterministic behavior of process execution; [4.5] Reproducibility.

Main drivers: End users of automotive and electrical industry and respective system integrators.

REQ 7. Acceptance of existing boundaries

As a consequence of REQ 5, existing boundaries of manufacturing systems must be accepted by the novel system as well. This effects the hierarchical separation of the traditional manufacturing systems in ERP, MES and shop-floor level and the clear separation of responsibilities for plant components such as machines or plant sections by personnel. Furthermore, a clear identification and assignment of errors, malfunctions, etc. to the respective equipment must be possible.

Reference to: [2.1] Conservation and interlinking of knowhow and expertise; [2.2] Consolidation of distributed demands; [4.1] Manufacturing organization; [4.3] Segregation of responsibilities in shop-floor organization.

Main drivers: Automotive end users of and respective system integrators.

REQ 8. Knowledge-based manufacturing: Knowledge on processes and methods should be used and re-used on all levels

Knowledge is commonly seen a one of the main drivers for future manufacturing systems. Knowledge on processes and methods can bring additional benefits for OEM's, system integrators and component vendors. For that reason, storing, accessing, (re-) using and spreading of knowledge is of great importance. Furthermore, distributed know-how must be interlinked and consolidated in order to gain most benefit. On the other hand, knowledge must be encapsulated and must be protected from unauthorized access. It is important, that the management of expertise and knowhow is easy. Even smaller companies must be able to handle future manufacturing systems without high efforts.

Reference to: [2.1] Conservation and interlinking of knowhow and expertise; [2.2] Consolidation of distributed demands; [4.1] Manufacturing organization; [4.3] Segregation of responsibilities in shop-floor organization; [2.3] Recipe management.

Main drivers: All, however with different purposes: End users target on increasing the effectiveness of manufacturing while system integrators and component suppliers target on advanced business models based on service-oriented contracts.

REQ 9. Components shall be integrated and used without requiring specific knowledge

In order to increase the systems acceptance, an easy integration of components into the system must be possible. This is mainly related to the component's interfaces. Interfaces are required which allow users to access components data and functions easily without having specific knowledge on the component's structure. For that, components shall be implemented as black-boxes which encapsulate specific knowledge and do have commonly usable interfaces for accessing it.

Reference to: [4.3] Segregation of responsibilities in shop-floor organization; [1.3] Flexibility of the production resources.

Main drivers: End users of aerospace and automotive industry and respective system integrators. Partially also relevant for electrical industry.

REQ 10. Traceability: Comprehensive data aggregation

Data which is gained during production must be documented and assessed in each stage of production. Within the requirement gathering, special emphasis was on the quality data which is linked to the produced products. To do so, the nowadays separated (and often only partial) assessment of quality on process level must be replaced by a 100% quality assessment. Production equipment must be enabled to assess its process quality automatically in a comparable way and format. Furthermore, factory-wide quality value gathering and assessment mechanism must be implemented. Quality data shall also be used for re-scheduling the production flow and for the optimization of the production process.

Reference to: [3.1] Quality monitoring and assessment; [3.2] Comparability of quality values; [3.3] Relation of process and product quality; [3.4] Product quality tracking; [3.5] Product and equipment optimization.

Main drivers: End users of aerospace, automotive and electrical industry.

The result of the requirement gathering process is a list of 10 essential requirements which have to be fulfilled in order to guarantee a smooth take-up of a future manufacturing system in the industry. The requirements especially are related to the more traditional-oriented industries. This is particularly reflected in those requirements which are related to the factory organization described in Section 3.5. The responsible managers of those industries are in a dilemma: On the one hand the need for novel approaches and ideas in manufacturing has been identified in order to keep the competitive advantage to other markets and to meet the future requirements of the customers. On the other hand high investment costs, organization efforts as well as aspects related to human behavior and thinking hamper the introduction of radical new approaches in one single step.

For technical point of view, future manufacturing systems needs to be adaptable for changing product and boundary conditions. The requirements gathered in this area relate to the definitions of the various fields of flexibility provided by Chen&Adam (1991). The main emphasis is seen in the product flexibility in order to enable the effective production of products in small lot sizes. The trend to customization and individualism is expressed in various papers (e.g. see Westkämper 2011). Furthermore, the need for plug&produce implies to equip controls and devices with standard interfaces. Such equipment provides the basis for flexible and adaptable manufacturing systems. Workprogramms related to such plug&produce equipment have for example been issued by the European Commission (EC 2011). Important conclusions are also related to expertise and knowledge handling. The requirement gathering has clearly shown that the knowledge on the respective business areas is seen as the core competences of the companies involved. In order to be competitive on the market, equipment vendors are forced to adapt their business models. In future business, expertise and knowledge are commonly seen as the factors which bring the added valued (see also MANUFUTURE 2004).

3.7 Characterization of related industries

The design of the proposed architecture for task-driven manufacturing is derived from the requirements expressed by the representatives of the automotive, aerospace and electrical industries that participated in the XPRESS project. In order to find out whether the architecture also meets the needs of other industries, this section characterizes three industry sectors which relate to those mentioned above. The selected related industries are: a) additive manufacturing, b) heat treatment and c) sheet metal industries. For all of them, a short explanation and a description of the main properties by a characterization of their processes and the process flows are provided below. Later on, the various characteristics are used for a validation of the proposed architecture in Chapter 6.

Additive manufacturing

Additive processes are defined as processes which create the desired shape of a product by an incremental addition of material in a layer-by-layer fashion (Black *et al.* 2013). Specifically this means additive processes can be classified in the applications' rapid prototyping (RP), production of scale models, rapid tooling (RT) and direct-digital manufacturing (DDM). DDM allows for the production of products directly based on a computer file with no intervening tooling. As a result, one-of-a-kind and small batch production can be executed more economically (Black *et al.* 2013). As the manufacturing system proposed within this thesis deals with executing production processes and production flow, we concentrate further on DDM, which is seen as being essential for the series production of products. In this context, the management of various orders coming from multiple customers is also taken into account. The major adopters of DDM are currently in the aerospace (e.g. for air ducts), automotive (e.g. for instrument panel) and medical industries (e.g. for metal implants) (Wohlers 2009). In 2012, the direct part production had a share of 28.3% of the total product and service revenues in additive manufacturing worldwide (Wohlers 2013). The current challenges that need to be faced cover the needs of process control and closed-loop feedback systems in order to improve repeatability as well as in streamlining the process of organizing and ensuring data quality of hundreds or thousands of parts (Wohlers 2009). Latest developments include the embedding of components such as sensors or microchips (see e.g. Maier *et al.* 2013).

Heat treatment

Heat treatment is used for controlled heating and cooling of materials for the purpose of altering their structures or properties (Black *et al.* 2013). The focus of further discussion will be on heat treatment of materials in terms of this definition and will not consider processes in which heat treatment occurs as an incidental phase of other processes such as welding or hot forming. More than 90% of all heat treatment is performed on steel and other ferrous materials (Black *et al.* 2013). Temperature profiles are used in order to heat and to cool down the materials during the process and online process monitoring is used for visualization and for adaptive control during the process (see also Demig 2004). As heat treatment can consume significant amount of energy, the reduction of energy consumption is currently one of the goals deemed most important to the industry, followed by the reduction of processing time and reduction of emissions (Black *et al.* 2013). ICT methods are used to optimize the heat treatment process itself (e.g. see Mendikoa *et al.* 2013). In the context of this thesis, the challenge for a manufacturing control system is to optimize the equipment utilization and the lead time of production. To do so, parts and components which require similar heat treating need to be grouped. Those parts may come from different orders and suppliers and may differ in geometry.

Sheet metal industries

Within this industry section we concentrate on the sheet forming processes which include shearing/punching operations and bending processes. In contrast to the bulk processes, sheet forming involves plane stress loadings and lower forces for material forming (Black *et al.* 2013). In the traditional way, shearing is done in an order-related job organization. To complete an order, an operator loads an NC program to the shearing machine, selects the tools and loads the metal sheet. The punching process forms the target blank out of the metal sheet and leaves small beams to hold the blank in the metal sheet. After completion of the punching process, the operator unloads the sheet and loads the new sheet. The operator manually cuts the small beams to release the individual work pieces. This process is repeated until the order is completed. The work organization proves to be cost-efficient when manufacturing large quantities of the same work piece, but inefficient for small lot sizes. For small lot sizes, nesting is a potential solution. Nesting combines multiple orders that require the same type of sheet metal. As a result, a punch press can complete all of the parts in these orders as one job. Dynamic nesting goes one step beyond as it periodically considers the specific orders to be punched in the next period and creates customized nests for those orders. (Herrmann&Delalio 2001). After the punching operation, the blanks might be bent. For bending different machine setups are required as the bending process needs to be adapted to the material properties such as material ductility and thickness (Black *et al.* 2013). Furthermore, a proper set of the parameters is required as they effect the distortion of the material (Black *et al.* 2013) and its springback behavior (Khamt *et al.* 2012).

Table 3. Characterization of related industries.

Criteria	Additive manufacturing	Heat treatment	Sheet metal
Major processes	3D printing e.g. laser sintering or laser melting	Heating/cooling and transportation	Shearing/punching and bending
Process flow organization	Job shop	Flow shop	Flow shop
Lot sizes	1 to tens	1 to several hundred	1 to several hundred*
Cycle times	Minutes … hours	Minutes … hours	Minutes … hours*
Production strategy	Computer Integrated Manufacturing	Batch production	Batch production
Level of automation	High automation level. Manufacturing is usually done fully automated without any human interaction	High. Automatically controlled heating and cooling processes; Transportation is usually done automated	Medium. Major processes automated. Manual operations in machine feeding and material transport
ICT usage	Integrated / interconnected systems for CAD handling and production	ICT used for heating and cooling process optimization	NC programming; Complex ICT solutions usually not in use
Products	Small to medium products	Small to large products	Medium products

* According to Ollikainen&Varis (2005)

4 Delineation of the architecture

This chapter describes the design of a new architecture for flexible manufacturing systems which fulfills the requirements identified in Chapter 3. To do so, it combines relevant elements of existing approaches for flexible manufacturing systems described in Chapter 2.

PABADIS'PROMISE showed that agile and flexible systems can be integrated into the traditional three layer architecture of traditional oriented industries (PABADIS-PROMISE 2008). Having such an architecture, REQ 7 "*Acceptance of system boundaries*" is fulfilled. On shop-floor, stationary agents similar to the PABADIS'PROMISE approach will be introduced. By this, REQ 2 "*Reconfigurability*" can be fulfilled. Furthermore, specific features will be added in order to also take care on the requirements REQ 3 "*Learning ability*" and REQ 8 "*Knowledge-based manufacturing*".

Team coordinators will be introduced which are inspired by the supervising holons of ADACOR. Both are able to orchestrate underlying production entities in order to realize global optimization (see requirements REQ 4 "*Scalability and extensibility*" and REQ 1 "*Flexibility*"). In contrast to the ADACOR approach, the design allows only restricted dynamic composition of the teams. Very dynamic behavior and team composition would lead to non-deterministic system behavior which is forbidden by design (see REQ 6 "*Deterministic behavior*"). Team compositions and their dynamic behavior must be proven in advance of production in order to guarantee a deterministic system behavior. For that reason, dynamic team compositions are simulated in advance of the real manufacturing process.

On MES level, ADACOR uses Petri-Nets for modeling the dynamic behavior. Petri nets become very complex for large systems and are difficult to understand (Leitão 2004). For that reason, a workflow approach is proposed to model the dynamic behavior.

Finally, the architecture is also designed in such way to meet the requirements REQ 5 "*Re-use of existing systems*" and REQ 10 "*Traceability*" which have not fully been addressed in the system design of existing architectures.

4.1 Overall methodology

The proposed architecture to overcome the gap between the ambitions in flexibility and real existing implementations is based on a bottom-up approach. It is designed to fit into the traditional three layer structure of factory organization.

The starting point of the bottom-up approach is on shop-floor level by a new concept for intelligent production equipment. Based on this, improved communication and routing capabilities on Manufacturing Execution System (MES) level as well as enhanced planning and optimization opportunities on Enterprise Resource Planning (ERP) level will be reached. Fig. 18 illustrates this approach.

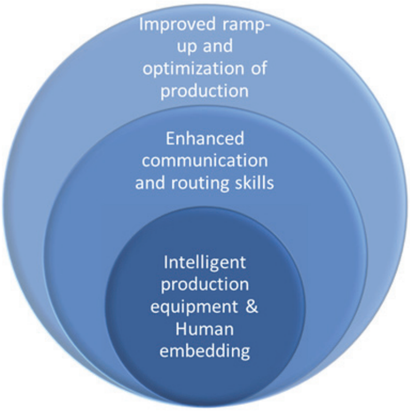


Fig. 18. Three phase bottom-up approach for step-by-step implementation.

From an abstract point of view, production entities in each level are supposed to perform an activity by the interpretation of input information and by subsequently deriving of the proper actions. After activity completion, the entities issue relevant information on the performance of the activity. In order to obtain such a behavior, the production entities have to contain the respective knowledge for performing the activities based on the input information. The design of the architecture needs to address this by bringing and exploiting knowledge and expertise in each level of the traditional manufacturing levels. Fig. 19 illustrates the methodology on an abstract level.

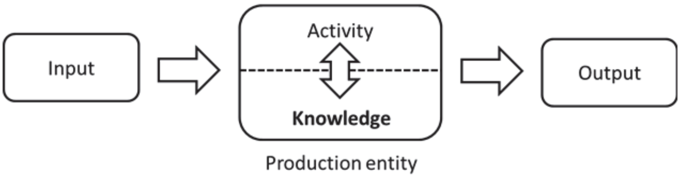


Fig. 19. Abstract scheme of the transformation process from input information to output information by a production entity using its specific knowledge.

The production entity performs an activity on the basis of input information and its intrinsic knowledge. After the activity is done, the production entity compiles and issues output information. In the next sections the overall methodology is described for each level in manufacturing. Special attention is paid to the aspects of knowledge which is required to perform the related activity.

4.2 Concepts for shop-floor organization

4.2.1 Intelligent production equipment

The outcome of the requirement gathering phase regarding the capabilities and properties of the equipment for future flexible manufacturing systems indicates that the exploitation and consolidation of knowledge on processes is required (REQ 8 "*Knowledge-based manufacturing*"). At the same time, the interaction with equipment must be very easy in that sense, that the access to the intrinsic knowledge is abstract and not related to the process (see REQ 9 "*Easy integration*"). The latter will enable equipment integrators and end users to integrate and to setup the devices easily without the need to have specific knowledge on the process or the internal functionality of the devices. The encapsulation of knowledge will also enable the equipment vendor for saving his knowhow and for the extension of his business models (see also Griffin 2009). In addition, re-use of equipment is required in order to improve the acceptance of the approach in the industry (see REQ 5 "*Re-use of existing systems*").

As a consequence, the architecture proposes to enhance conventional devices and controls with capabilities for knowledge handling and knowledge encapsulation in order to create a new kind of flexible and intelligent production equipment[2]. In order to enable flexible behavior in case of changing demands in production and to also enable an easy use of the equipment without specific knowledge, it is proposed that the equipment receives only instructions on the task to be performed. The equipment shall decide by itself how an activity shall be performed. In other words: Equipment receives only the "*What*" to be done, but not the "*How*". How to optimally perform a job is the intrinsic knowledge of the equipment. This is a significant contrast to conventional equipment which

---

[2] In order to increase the readability of this thesis, the new kind of intelligent and flexible equipment which builds the core of the proposed approach is from now on simply termed as "equipment". In cases where today's conventional equipment is meant, this is termed as "conventional equipment".

receives recipes including the parameters for executing a task. If a task changes e.g. due to changed geometries or materials of the assembly, re-programming in in a task-driven environment in the conventional sense is not required. Only the task description needs to be modified and the equipment adapts itself for executing the new task.

Chryssolouris (2006) showed the relation of flexibility and quality. For that he describes a tetrahedron of competing objectives of manufacturing systems (Fig. 20). If flexibility is introduced, the manufacturing process quality is influenced.

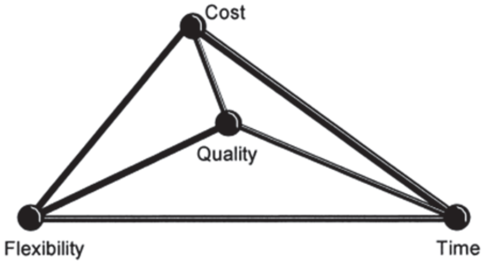


Fig. 20. Manufacturing tetrahedron (Chryssolouris 2006, with kind permission from Springer Science and BusinessMedia).

As the behavior of the equipment shall be most flexible, an assessment of the result after each operation is done. For that reason, the equipment shall be equipped with features for assessing the result of its activities. In usual cases, the result assessment reflects the quality of the result. The quality result is issued by the equipment in a document which can be interpreted by other entities of the factory for further processing. Fig. 21 illustrates the approach[3].

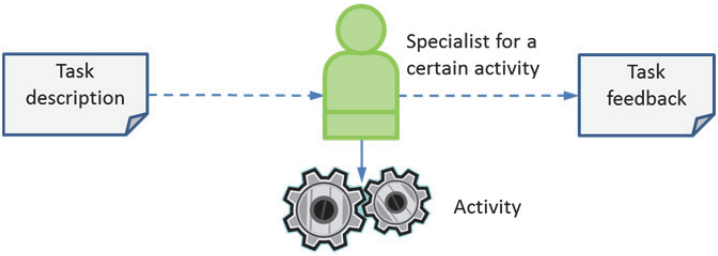


Fig. 21. Principle mechanism of an intelligent task-driven device operation.

---

[3] Because of the high similarity to human experts, the task-driven devices are illustrated in this chapter in a shape similar to the human body.

The difference of the conventional and the new approach is illustrated by an example for joining equipment: Nowadays vendors of welding control units usually deliver solely the devices. The user of a welding device is in charge of programming the device manually by setting process parameters such as current level or welding time in order to obtain a sufficient quality for the join. This requires specific knowhow and training of the welding process behavior for users. This holds true for all other processes and devices in a manufacturing system.

The novel approach proposes to encapsulate process knowledge as an intrinsic part of each device or control. This means, that the equipment vendors are in charge of gathering and maintaining process knowhow and encapsulating it in their devices. After the physical integration of the device in the manufacturing system (mounting, cabling, etc.), the device is immediately ready to use. The command for performing a welding job is just the description of the job properties such as the material types, boundary conditions and the expected quality. The device is able to perform the welding job only by this task description. For the communication of the equipment during processing of the instructions, an event mechanism is implemented. While the usual operation is a synchronous communication mechanism (receive instruction → perform activity → issue result), the communication via events is asynchronous. This allows sending signals to the equipment at any time and the equipment is able to send signals to the outside e.g. in order to provide status information.

Table 4 summarizes the set of information which is handled by a task-driven device.

Table 4. Summary of the core information of a task-driven device for the performance of an activity

Information	Content
Input	Description of the task to be performed; Description of the goal of the task, Relevant boundary conditions
Output	Result of the execution of the task described in the input
Activity	Task interpretation; Task execution; Task result assessment
Knowledge	Knowledge on the processes for task performance; Knowledge on the own capabilities

In order to perform complex tasks within a factory, various teams composed of different equipment are required. Each member of a team is an expert for performing a specific subtask. If for example the goal of the task is to assemble two subassemblies to a more complex assembly, there might be equipment for handling and fixing the two subassemblies and one device for performing the joining job. To do so, the team members must communicate in order to fulfill the task goal. Fig. 22 shows the approach for a general structure of team work for task-driven devices.

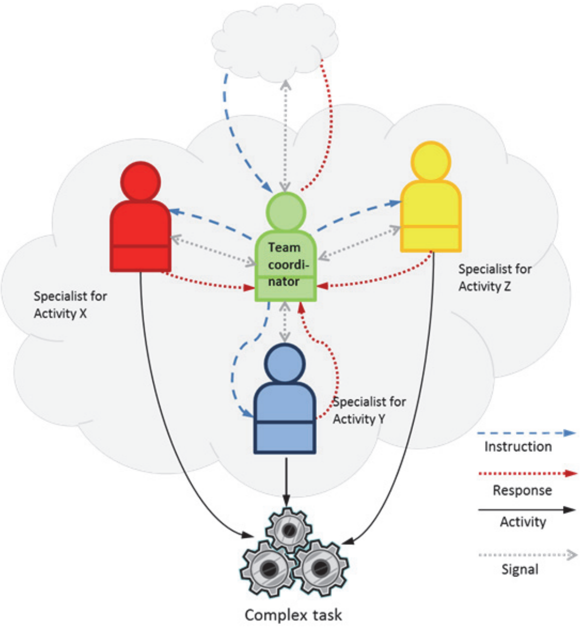


Fig. 22. Teamwork of task-driven devices is orchestrated by a team coordinator.

For the coordination of the team members a dedicated *team coordinator* is established. Team coordinators can have different roles: A coordination role which is – similar to a foreman or team leader – not performing a physical task, but is solely responsible for distributing instructions, for synchronization and for supervising the performance and the results of the other team members underneath by the assessment of their responses and an executive role for performing physical tasks. The team coordinator is the only interface from the team to the outside. All communication related to instructions and response is done via the team coordinator.

The content and scope of the instructions which are communicated to the team members depend on their role in the team. The team leader, who obtains his instructions from outside receives information on the job to be done by the team. He is responsible to coordinate his team in such way, that the job is done properly. The team members receive their instructions from the team coordinator on dedicated subtasks. For building an assembly *AB* composed of two subassemblies *A* and *B* for example, the team coordinator might receive the instruction “Build assembly *AB*” from outside and issues the instructions “Put subassembly *A* to position (*a*, *b*, *c*)” to team member *X*, “Fix subassembly *A* and *B*” to team member *Y* and “Join subassembly *A* and *B*” to team member *Z*. This example also shows that the level of detail in the instructions is higher for the team members than for the team coordinator or vice versa, the team coordinator receives more abstract information than the team members.

For the synchronization of subtasks performed by the team members, two different mechanisms are available. Subtasks which are not time-critical can be synchronized by the team coordinator. The coordinator issues the set of instructions to each team member in the sequence of operation. After the performance of the subtask, the team member issues the task result to the coordinator. By this, all communication is done via the coordinator without a direct communication mechanism between the team members.

For time-critical jobs, a fast direct communication mechanism is required. However, the direct communication must be designed in such a way, that the independency of the knowledge domains of each team member is guaranteed. Otherwise the exchangeability of equipment might be hampered. In addition to that, no negotiation mechanism shall be used in order to avoid non-deterministic behavior. Details on synchronization mechanisms are provided in Section 5.2.5.

As described above, the team coordinator receives the relevant instruction for task performing from outside and orchestrates his team accordingly. However, the team must not be composed by static entities. In order to be flexible on varying tasks and boundary conditions, a dynamic allocation of equipment is required. This enables for the adaptation on unexpected production volumes by the temporary allocation of redundant equipment as well as for the exchange of equipment e.g. in case of changing tasks. Naturally, such mechanisms do have restrictions in the manufacturing phase due to limited resources and fixed mounted equipment. For that reason only limited dynamic allocation is possible during manufacturing, especially in the industry branches under investigation. On the other hand, this mechanism is the basis of finding a proper setup of the production process during the design and optimization phase. Further details on this aspect are described in Section 4.4.

For very complex jobs which are composed of several subtasks the composition of only one team might not be enough. Especially in such cases where tasks must be performed in parallel, more sophisticated equipment orchestration is required. For that reason and to also improve the scalability of the entire system (see REQ 4 "*Scalability and extensibility*"), the final architecture design allows for building of teams which are composed of several sub-teams. All team coordinators from the sub-teams are coordinated by a superordinate coordinator. This leads to a hierarchical structure of equipment organization and communication.

4.2.2 Learning process and equipment networks

For performing the tasks only by receiving a task description, the equipment requires the capability to store and to manage the know-how on the respective process. In case of a welding control unit for example, the task description would contain information on the materials to be welded as well as the desired quality of the welding joint. The know-how of such a controller is the settings for performing the welding job such as welding current, welding time, etc. A set of such information which is related to a potential job description is called a *method*. The more methods are available, the higher is the probability for fulfilling the task goals properly. For the generation of methods, several possibilities do exist. Fig. 23 provides an overview of various options for method generation.

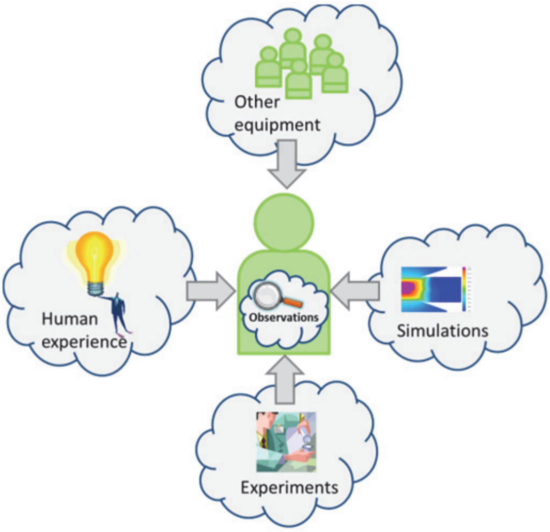


Fig. 23. Five options for method generation of a task-driven device.

The first option is auto-generation of new knowledge by the observation and subsequent analysis of the manufacturing process and its environment. This can be achieved by monitoring the process conditions and signals with the help of sensors and the subsequent derivation of relevant information on the setup, new knowledge can be generated inline. For process data interpretation several potential technologies do exists such as data mining, statistical process control (SPC) or unsupervised learning methods.

The second option is the elicitation of knowledge of a human expert. Experts of a specific area in production or of a certain process are often workers, operators or maintenance personal who are daily operating with the respective machines and/or processes. Often, these people do have an intuition on how to setup a certain process optimally under the given conditions. In practice this method is often used when the basic configuration for setting up the respective process are finalized and only fine-tuning of parameters has to be done. However, the formal specification of the (intrinsic) knowledge is difficult or even not possible which can be seen as a potential drawback.

The next option is to derive potential methods from physical experiments. If for example methods for performing a joining process such as riveting or welding shall be generated, a set of tests under varying conditions can be done. This option is often done in such cases, where new materials or processes are involved and no pre-experience on the process is available. The main advantage of this option is the systematic way of method generation. After performing the experiments, proper settings for the process under investigation are available. In addition to that, the experiments often include quality tests. By this, the method (the set of information required for process execution) already is “proved” by physical process execution and subsequent quality tests. The main disadvantage of this method is the potential high effort for performing the physical tests. Machines and devices as well as human experts must be available to carry out the tests. Furthermore, the quality assessment must often be done manually which again increases the efforts (e.g. in case of welding experiments destructive tests are required in order to measure the quality of the joint).

The fourth option for method generation is the performance of simulations. This option is – in principle – very similar to the performance of physical experiments. Most simulation tools available on the market implement a physical model of the process under investigation. According to the settings of the simulation base parameters, the process is virtually executed. At the end of the simulation process, the settings for the performance of a dedicated production task are available and can be added to the equipment's internal knowledge system. However, the quality of the generated method highly depends on the quality and reliability of the simulation tool. Often, a physical experiment is done subsequently to a simulation in order to prove the simulation result.

Fig. 24. Learning in a network of task-driven devices by interconnecting their specific knowledge.

The last option for method generation is to receive the method from other equipment. This is done by connecting equipment of the same type via a network and by subsequent synchronizing the stored know-how. Fig. 24 illustrates this approach. By this, the latest information on how to optimally perform a task under the defined boundary conditions is available immediately. The knowledge is automatically updated by the learning cycle whenever new methods are found to satisfy new needs encountered on site or if more efficient methods are found for existing processes. By this, the equipment of the same type spread over all connected factories, is forming a permanently learning network with respect to their specific process and is having always the latest knowledge available on site. In order to meet REQ 7 "*Acceptance of system boundaries*" and in particular the aspects on the separation of responsibilities on shop-floor, process knowledge is communicated in the same knowledge domain only. No domain bridging is foreseen. Furthermore, restrictions on knowledge exchange might be required if equipment (and thus, knowledge) from different vendors is used.

The various options for generating methods for process execution directly target the aim to build a system which is capable for constant knowledge creation and exploitation (see REQ 3 “*Learning ability*” and REQ 8 “*Knowledge-based manufacturing*”).

4.2.3 Human embedding

There is no doubt that the most intelligent and flexible entity in manufacturing is the human. This fact is also expressed in REQ 2 “*Reconfigurability*”. For that reason, the proposed architecture also covers the integration of humans in semi-automated manufacturing environments. The overall concept for embedding humans is called the *Manufactronic Workplace* (Peschl 2012b). Because of the similarities between the new kind of equipment and humans, the embedding of humans can be done with reasonable efforts. In order to perform a job, humans also receive the instructions and the boundary conditions which have to be taken into account. Humans are also able to provide a result on the performed task. This shows that the “interface” of human and intelligent production equipment is very similar. However, there are some differences and important boundary conditions which must be noted. In order to profit most of human’s expertise and knowledge, the task execution should not be as fixed as for machines. In a human-oriented workplace some freedom for acts and decisions on how to fulfill a given task should be given, e.g. in which sequence different working steps will be done. This can also contribute to an improvement of efficiency by finding best practice for an already established task. Another aspect belongs to the interpretation of the instructions for performing a task. In case of production equipment, the instructions must be machine-readable while for humans the machine-readable instructions must be converted in a worker understandable and executable method. Proper methods for displaying the task content have to be present. Online help and guidelines including advanced presentation technologies such as movies, 3D animations, etc. should be available. For the inspection of the quality of the performed task and for the safe interaction of workers and machines, several technologies for worker behavior monitoring can be used (see e.g. Hartmann *et al.* 2010 or Koskimäki *et al.* 2013). However, the measures for those activities have to be balanced very well. Worker monitoring must not be abused for total surveillance and control of the worker. Legal aspects for such measures which are potentially country-specific have to be taken into account.

4.3 Communication and routing on MES level

After the description of the intelligent equipment which serves as the basis of the approach we now can go a step further. The next level under investigation in the overall methodology is the MES layer. The requirement gathering process showed that enhanced communication skills of equipment and routing capabilities of the production flow are needed. Basically, there are two main goals for such capabilities: a) to gather process information such as quality values for each production step and to relate this data to the respective product and b) to enable the production flow for dynamic routing in case of e.g. a detection of insufficient quality or bottlenecks (see REQ "*REQ 10. Traceability:* Comprehensive data aggregation" on page 62). Those features target the interaction of controls and devices along a production line or even within the whole factory. As shown in Section 4.2.1, equipment on the lowest level of the production process can be grouped in teams. Such teams form similar to conventional factory organization the work units which are dedicated to manufacture complex assemblies. The arrangements of all work units within a factory build the factory layout. The overall manufacturing process for a certain product in the factory is then a sequence of commands to the team coordinators what to do in the respective unit combined with a set of data defining the respective boundary conditions of a processing step. Fig. 25 illustrates the normal operation of production flow.

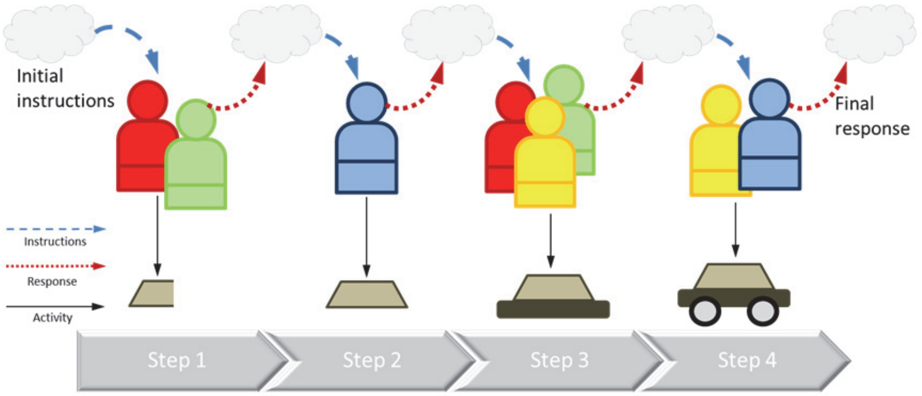
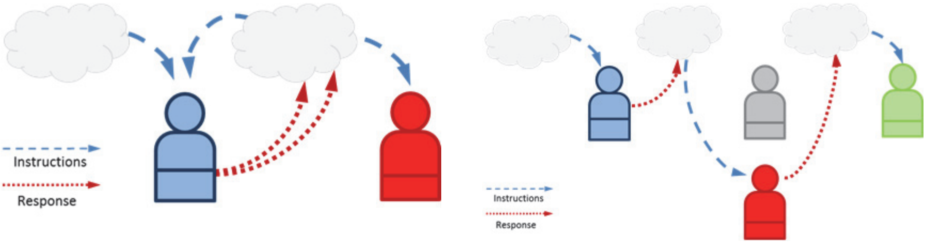


Fig. 25. Normal operation of production flow: Task descriptions are processed from production step to production step. In each step, response data is compiled and processed to the next step. Task descriptions and response data are accompanying the physical products along the production chain.

After receiving the set of initial instructions, the first team performs its work according to the description of the tasks included in the instruction set for this team. When the work is finalized, the team compiles the response data which includes the quality of the tasks performed in normal operation or error information in case of a malfunction. Both data sets will be transferred to the subsequent device or team in the production sequence in parallel to the material flow. At the end of the production sequence, the final response data is available which contains the entire information on the operation and quality of each production step.



(a) (b)

Fig. 26. Examples for dynamic routing a) repeating of a job within the same unit and b) re-routing of the workflow to an alternative unit.

In order to react flexibly on the outcome of the process steps, an assessment of the response data after each step is done. According to the assessment results the workflow is adapted. Potential reactions of the workflow are (i) to repeat a production step within the same work unit (ii) to re-schedule the workflow e.g. to a re-work station or (iii) to drop out the product e.g. in case of totally insufficient quality (see Fig. 26). To do so, the set of instructions is adapted dynamically based on the assessment of the response data. The set of instructions, gathering and assessment of the response data as well as the dynamic adaptation of the instructions according to the assessment results are done by (software) entities localized at the MES level. Table 5 summarizes the information which is relevant to the workflow execution on MES level.

Table 5. Information and its content for the workflow execution of MES level.

Information	Content
Input	Electronic Bill of processes (eBOP) for the execution of all production steps; Task feedback of the previous process steps
Output	Modified sequence and/or task descriptions for the next process steps
Activity	Deriving of routing alternatives depending on the feedback of previous production steps; Supply of the subsequent production units with relevant task information
Knowledge	Workflow routing; Quality assessment and consolidation of quality results; Strategies for contingency plan development, e.g. load balancing

4.4 Generation of workflows on ERP level

As the reader now is familiar with the proposed concepts for intelligent production equipment and with the principles for overall quality assessment and dynamic routing, this section describes the last missing fragment of the architecture: How the instructions for the various devices and controls are created and how the architecture contributes to the need for fast ramp-up and for optimization of production.

For each product type which shall be produced, a different equipment orchestration and different setups for the single devices might be required. Taking this into account, the consequence is to create a dedicated set of instructions for each product type. Furthermore, also different setups of production equipment for the same product type might be useful, e.g. if the optimization goal of the entire production setup shall be flexible. The compilation of the instruction sets for the products and product variants is done “offline” in advance or in parallel of the production process. Fig. 27 shows the approach.

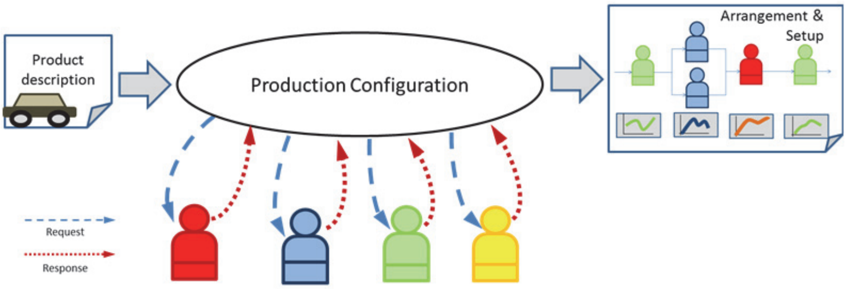


Fig. 27. Workflow generation on ERP level with the help of the product description.

The generation of workflows follows the general approach for a transformation of input information to the desired output by the usage of knowledge and expertise. The input of the configuration process is the description of the respective product or product variant. Furthermore, some other information such as the description of the optimization goal for the entire production setup, restriction of the available equipment, etc. needs to be specified. The output of the configuration process is the set of instructions for all equipment as well as the process sequence for production (Electronic Bill of Processes, eBOP). The transformation of input to output data is done by using specific knowledge on the processes required underneath as well as on the expertise for the simulation, optimization and cost estimation of various arrangements and orchestration of equipment. The configuration process works as follows: Starting with a meaningful production setup and set of instructions, all equipment is “asked” if it is able to perform the job and which job results can be expected. The equipment responses with the requested information. The communication scheme is similar to the one used for task execution described in Section 4.2.1. However, there are two significant differences: Firstly the equipment does not execute the task, but performs a virtual task execution and result assessment. To do so equipment must be equipped with capabilities for emulation and simulation[4]. Secondly the physical hardware related part of the equipment is not necessarily required. As the capabilities for simulation and emulation are usually implemented in software only, a virtual instance of the equipment can be used to perform the job. After the responses of all issued tasks are gathered, the production configuration performs an assessment of the setup by calculating the costs function related to the desired optimization targets (see also REQ 2 “*Reconfigurability: Optimization goal flexibility*”). Depending on the result of the calculations, the (virtual) setup will be rearranged and the next simulation iteration can be issued. This sequence will be repeated until an optimal setup is found. On an abstract point of view, the generation of the eBOP is very similar to the method generation of equipment described in Section 4.2.2. The result of the simulation process can be interpreted as the generation of a method for optimal production configuration with respect to the given boundary conditions. The collectivity of production setup builds the intrinsic knowledge of the production configuration. Similar to the generation of methods for process execution on shop-floor and the generation of workflows on

---

[4] *Emulation* in this context is the execution of an existing method without performing the physical process. *Simulation* is the generation of a new method. See Section 5.2.2 for a detailled description.

MES level, the generation of production setup on ERP level is a part of the architecture's learning network to always represent the latest methods. Table 6 summarizes the relevant data for workflow generation on ERP level:

Table 6. Information relevant for the generation of workflows on ERP level.

Information	Content
Input	Product description; Optimization target
Output	eBOP including the task descriptions for all process steps
Activity	Performance of distributed and parallel simulations
Knowledge	Process knowledge of the involved (virtual) devices; Knowledge on optimal production setup under various boundary conditions

4.5 Manufactrons: A definition

Within the last three chapters, the general ideas of task-driven manufacturing have been described. The basis of the approach is built on intelligent which are capable to perform manufacturing tasks by receiving a description of the task. Such task-driven devices are named *Manufactrons*. The definition of the Manufactrons is done based on four different views (Fig. 28).

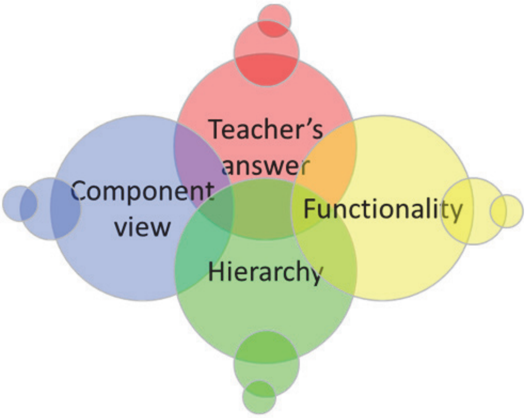


Fig. 28. The four views on the definition of Manufactrons (Peschl *et al.* 2011, published by permission of OmniaScience).

The *Teacher's answer* summarizes the definition into one single and very generic phrase:

> *"A Manufactron is a self-contained entity, which is encapsulating expertise and functionality and interacts with its environment by the exchange of standardized, synchronous messages."*[5]

Self-contained means, that the Manufactron can be integrated into a manufacturing system just by plug&produce. All required functionalities for this behavior shall be integrated in a Manufactron and no additional integration efforts shall be required. Manufactrons encapsulate expert knowledge and provide an interface for accessing this expertise without specific knowledge. This behavior is very similar to the paradigm of object-oriented programming (OOP) (see e.g. Booch 2007). Manufactrons are interacting with their environment and are building networks of Manufactrons. To do so, they exchange messages which shall comply to a standard. Those messages shall be synchronized which means that every message to a Manufactron shall result in an answer before the Manufactron is able to process the next message. According to the definition provided in Section 2.2, Manufactrons are agents with special features and skills.

The *component view* lists the several components of which a Manufactron is composed of:

- Components to send, receive and analyze Task Description Documents and Quality Result Documents[6];
- Components to configure it's logical position (in hierarchies) and behavior in a standardized way;
- A knowledge base to store 'method data‘ (locally or remote);
- An 'unsharp' knowledge retrieval mechanism;
- A local database for process data and quality data (including aggregation of data and calculation of quality data);
- Interfaces to interact with dependent Manufactrons, connected hardware or connected software components.

The *functionality view* lists the functionalities of a Manufactron. According to this, a Manufactron is able to

---

[5] Dr. Norbert Link published in Peschl *et al.* (2011).

[6] The *Task Description Documents* and *Quality Result Documents* are defined in Section 5.2.

- Perform a task autonomously by using only dependent Manufactrons or optionally connected hardware and software;
- Analyze the Task Description Documents for task goal and boundary conditions;
- Analyze and integrate available data to a Quality Result Documents;
- Understand standardized configuration data;
- Describe itself via a self-description document;
- Understand the self-description of other Manufactrons;
- Store and lookup method details in a knowledge base;
- Automatically learn new methods from data and expert knowledge;
- Perform a functional emulation of its own;
- Provide a (list of) best fitting method(s) to a given task description;
- Optionally share knowledge by the use of the method knowledge base;
- Optionally do a simulation in order to determine method details;
- Optionally generate sub-Task Description Documents and analyses sub-Quality Result Documents;
- Optionally synchronize with other Manufactrons for real-time tasks.

The *hierarchy view* defines the entities within the layers of a manufacturing system which are required in order to realize the *Manufactronic Factory* (Fig. 29).

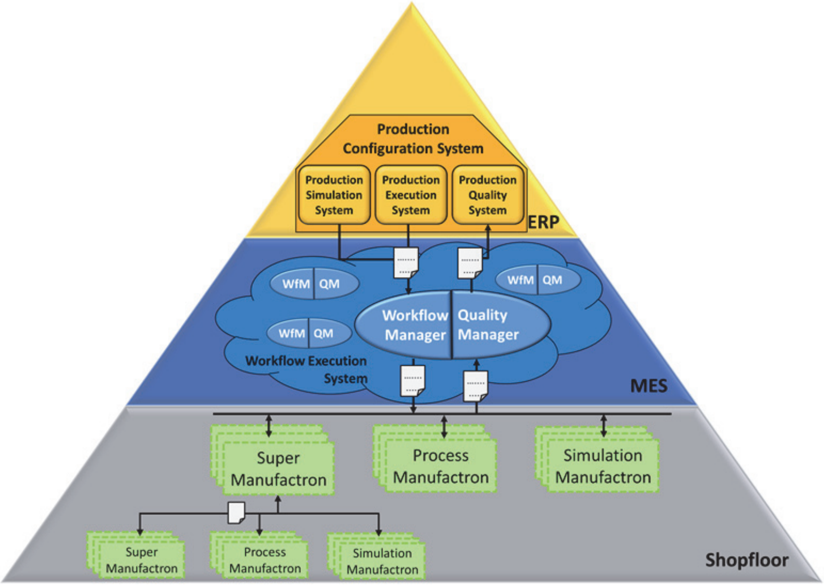


Fig. 29. Hierarchical structure of the Manufactronic Factory.

The Manufactrons are residing on shop-floor level. They can be either implemented as single instances or build hierarchies. Manufactrons are classified in the three types (i.) *Super Manufactrons*, (ii.) *Process Manufactrons* and (iii.) *Simulation Manufactrons*. Super Manufactrons are building hierarchies with other Manufactrons and are explained in detail in Section 5.2.1. Simulation Manufactrons perform dedicated simulation jobs. They are most useful to interface complex tools for simulating e.g. entire factory setups (Peschl *et al.* 2012a). Process Manufactrons represent the machines and devices on shop-floor. The taxonomy of the Process Manufactrons is illustrated in Fig. 30:

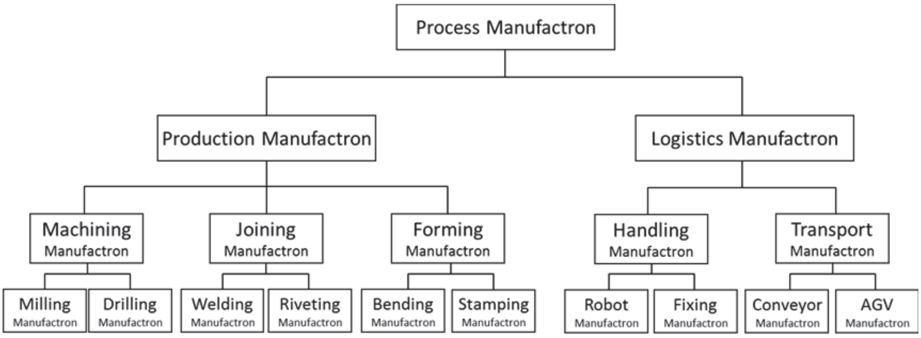


Fig. 30. Taxonomy of Process Manufactrons.

The workflow mechanisms for Manufactron orchestration are implemented on MES level. In addition, also quality data collecting is done. For those purposes, so-called *Workflow Managers* and *Quality Managers* are introduced. The mechanisms on MES level are described in Section 5.3.

On ERP level, production planning, production execution and production quality assessment is done. For that purposes, the so-called *Production Configuration System* with its three sub-systems *Production Simulation System*, *Production Execution System* and Production Quality *System* is located. The Production Simulation System is supposed to simulate production processes in order to generate workflow templates. It is responsible for issuing simulation jobs to the Manufactrons underneath on shop-floor level and for receiving and assessing their simulation results (Peschl *et al.* 2012a). The Production Quality System is responsible for storing and retrieving the quality results in form of the Quality Result Documents (Peschl & Hoffmeister 2011). The Production Execution System is supposed to generate instances of the Workflow Managers for each product to be produced. The generation of Workflow Managers is described in Section 5.3.1.

4.6 Discussion

In this chapter a novel architecture for flexible manufacturing systems has been described. The proposed architecture can be seen as a combination of several aspects which has been addressed by other approaches for manufacturing system and additions for the management of knowledge. The proposed architecture is tailored for the needs and requirements for such industries which are facing by the same challenges and needs identified in Chapter 3.

The major novel aspects of the proposed architecture with respect to the state of the art are related to the improvements on shop floor level. Traditional equipment relies on fixed setup and programming sequences for in order to gain proper process execution. This means that the users of the equipment needs (deep) knowhow on the respective process. Manufactrons enhance traditional equipment with the capability to understand task descriptions and to derive proper methods automatically for task execution. By this, the operation becomes easier as the users do only have to describe the task without requiring knowledge on the process itself. The transition from fixed equipment setups to flexible task-oriented process execution denotes a paradigm change in the way of equipment programming. Task descriptions are also used for accessing the Manufactron’s encapsulated skills on the execution, emulation and simulation of tasks as well as for the integration of humans. By this a standardized common interface and communication scheme are established which supports the easy integration of Manufactron in a factory. As the relevant process knowledge is an intrinsic component of each Manufactron in a manufacturing system, the knowledge is distributed in the network of Manufactrons. As soon as a new method for a task description is generated locally via the Manufactron’s learning capabilities, this knowledge is immediately available for all other Manufactrons in the network within the same knowledge domain. In comparison to centralized approaches as e.g. discussed in Eigenbrod (2008), the distributed approach provides a better robustness.

The need for hierarchical entities for equipment aggregation and orchestration has been investigated in several projects (cf. Leitão 2004). The Super Manufactrons go a step beyond as they implement another class of task-driven device. The knowledge on how underlying Manufactrons are orchestrated in order to fulfill the desired task goals is stored as the intrinsic knowledge of the Super Manufactron. Super Manufactrons might for example be used as the coordinators for the equipment within one production cell. Their intrinsic knowhow comprise the knowledge on the orchestration of the equipment for producing a (sub-) assembly in the production cells. Having this, the cell integration into a production line from IT point of view is easy: For the integration solely the task descriptions to build (sub) assemblies need to be defined and forwarded to the Super Manufactrons. The equipment orchestration is done by the Super Manufactrons without further manual equipment setup.

The architectures' features on MES and ERP level assist the task-driven approach on shop floor. On MES level, the most important aspect is the combination of quality information coming from the shop floor Manufactrons and the routing of the production flow. The demand of flexible process execution on shop floor level emerge the need of having full quality control of the process outcome. The need of routing capabilities emerge dynamic decisions of the production flow based on the quality outcome of the different production steps. On the other hand, the requirement for deterministic system behavior leads to strong restrictions of those features, at least for the industries under investigation within this thesis (see Section 3.6). In other industrial environments, which allow for a more dynamic production execution, other and more elaborated approaches are allowed and required. Several of them have been shown and discussed in Chapter 2. For that reason, the features for dynamic routing of the proposed architecture is also covered by current state of the art approaches. Nevertheless, the outcome of the requirement gathering process clearly showed that a proper mechanism for the handling of data feedback from the shop floor in a standardized and easy to use way is an important feature of an architecture of a novel manufacturing system. Such a mechanism is therefore covered in the architecture proposed within this thesis.

For other factory setups which allow for a more dynamic behavior on MES and ERP level, the implementation of an instance of the PROSA reference architecture with task-driven features might be useful. In such an environment, the Manufactrons would be a special implementation of the resource holons covering the physical hardware and extending it with task-driven capabilities. In order to guarantee the fast response on task requests, task descriptions should be as short as possible to minimize the task description interpretation time. Furthermore, task scheduling should be implemented in the Manufactrons, meaning that a Manufactron shall be enabled to receive multiple, subsequent task descriptions even if the Manufactron is currently processing a task. By this feature, minimum delay between the receipt of a task description and the respective task execution is guaranteed. The workflow component which actually is composed of product and routing information could be replaced by order holons and product holons.

On ERP level, the three systems Production Simulation System for distributed simulation and workflow template generation, the Production Execution System for workflow instance creation and the Production Quality System for quality data handling do mainly play a supporting role for the architecture. Distributed simulation with the help of the Manufactron's emulation and simulation capabilities are important for the acceptance of the industry as it strongly eases the handling with vendor-specific task descriptions. For that reason, those mechanisms are discussed more in detail in Section 5.3.1. The other features on ERP level within the proposed architecture are covered by state of the art technologies and are therefore not further elaborated in this thesis. For the acceptance of the industry, however, a crucial point is the potential for interfacing existing (commercial) solutions for ERP systems. The more easily the connection to existing solutions is the better acceptance will be reached. In order to obtain an estimation on the implementation efforts, investigation on different solutions for interfacing commercial ERP systems have been done by two XPRESS partners for two different applications. The first application was the quality inspection of the car body painting. In this application, the painting of the entire car body is scanned with cameras mounted on robots and any flaws such as inclusions or color defects are identified by subsequent image processing. The major information coming from the commercial ERP for workflow generation was the car type (for robot path generation), the color and the related quality level to be reached. The data fed back to the ERP system was the data on the quality[7]. The other application is a setup of a virtual production line in the photovoltaic industry. The data from the ERP system mainly includes product information and batch sizes. The feedback data consists of the quality items gathered within the different production steps. A more detailed description on this application is provided in Section 6.1.4. A possible way for interfacing the respective ERP systems is to use the ERP's application programming interface (API) which allows the programming of customized software modules within the ERP system. Even if most of the ERP systems provide such an API, this solution has a number of disadvantages: The respective software module to be implemented is tailored to the respective ERP system and thus, cannot be used generically. Furthermore, version conflicts might occur as ERP systems are usually updated in frequent intervals. In order to avoid such problems and in order to keep the efforts for the implementation low, both XPRESS partners decided on a different solution based on text-based import and export functionalities. Most of the ERP systems provide configurable export of data in a text format, e.g. xml and allows also for importing such data from text files. Using this functionality, no modification of the

---

[7] Due to confidentially reasons a more detailed description of the application cannot be provided in this thesis.

ERP system itself is required. The respective import and export algorithms within the Production Configuration Systems can be adapted easily for different ERP systems. In addition to the file-based import and export, most of today's ERP systems provide web-based services (e.g. REST interfaces) for standardized data exchange.

By giving the different entities of a manufacturing system the capability to understand task descriptions and to derive proper methods which will be executed, the system is enabled to react dynamically on changing demands in production (see REQ 2 "*Reconfigurability*"). In addition to this, this approach allows vendors to integrate their specific process knowhow as an intrinsic element of their devices and controls. By this step, vendors are able to create new business models as they are now enabled to offer not only devices, but also the knowledge on the respective processes (see REQ 8 "*Knowledge-based manufacturing*"). The crucial and challenging point is the generation and storage of explicit process knowledge within the devices as well as the derivation of proper methods for an incoming task description. A detailed discussion with respect to this is done in Section 5.2.8.

Another advantage of the architecture is the possibility of a step-by-step implementation in the factory: With the introduction of a new kind of intelligent production equipment on shop-floor, benefits can already be reached even if the overall architecture is not implemented on MES and ERP level. For example, quality assessment on process level can be introduced or recipe management can be reduced by task-driven execution in dedicated sectors of production. In addition to that, a mixed operation of intelligent and conventional equipment on shop-floor is possible by wrapping conventional devices with Manufactronic shells and by adapting their communication scheme. Thus, existing equipment can be upgraded or replaced in accordance with the required needs and the available resources. Furthermore, the enhanced routing and communication skills including quality gathering and assessment mechanisms are independent from the implementation of the methodology on ERP level. The full capacity and benefit, of course, can be exploited if the architecture is fully implemented in all levels of the factory.

The design of the manufacturing system described above surely consists of some restrictions and limitations in comparison to the already existing approaches identified in the state of the art analysis. Most important to mention at this point is the fact, that the design of the manufacturing system is only *one* possible solution for satisfying the needs and requirement of the target industries. The design is the result of abduction, meaning that it can be concluded based on the requirements described in Chapter 3. However, there are potentially other architectures with alternative design which could also fulfil the requirements. Those alternative designs have not been discussed within this thesis.

A general potential weak point of the architecture is that the design relies on an early and high commitment of the constraints on flexibility in process execution. The decisions in flexible behavior the system can do during runtime are relatively limited as the alternatives need to be simulated in advance of the execution process. This especially holds true for the operations on MES and ERP level. However, the argumentation of the industry partners of constraining the system's flexibility in order to guarantee deterministic behavior does not necessarily impose such restrictions in a very early stage. Late commitment – as for example proposed in the PROSA reference architecture – does also allow for deterministic system behavior provided that the respective system implementation is taken this into account.

In addition to this general remark on the architecture design, there are a number of other limitations which need to be stated shortly. The restriction of the limited dynamic team composition has already been mentioned above. As said, this is introduced to guarantee production cycle times within a fairly fixed timeframe. For that same reason, negotiation between the production entities during production execution is not allowed. This is replaced by the "offline" generation of workflows on ERP level in advance of the production.

A limitation and potential crucial aspect is related to the knowledge exchange. Knowledge exchange between production entities is done solely within the respective knowledge domain. No knowledge sharing or exchange between domains (domain bridging) is foreseen. This limitation hampers the identification and usage of potential cross-domain knowledge. Furthermore, the same knowledge might be useful and required in different domains. In such cases data is stored redundantly causing all well-known issues of redundant data storage.

Finally, a crucial aspect has also been identified in the mechanism of task execution. The communication mechanism of task-driven production is designed as a synchronous communication protocol. A new task can only be executed, if the old task has been finalized and the result has been assessed and issued. Especially in case of huge and long-lasting tasks, the respective equipment is blocked during the task execution time. For that, huge tasks should be avoided and asynchronous communication mechanisms, e.g. for task interruption or state information gathering have to be implemented.

5 Realization

The goal of this chapter is the explanation of the core elements of the architecture described in the last chapter. Section 5.1 explains the approach for *task-driven manufacturing*. For that, the main aspects of this approach are discussed in detail. After starting with a description of the principles of task-driven manufacturing, the purpose, the structure and the content of the so-called Task Description Documents are described. Next, the approach for deriving methods for task-driven manufacturing and for quality assessment is illustrated. Finally, an example for task-driven manufacturing is provided. In Section 5.2 a model for *hierarchical organization and communication* is described. This model builds the second pillar of the proposed architecture. First, the organization of production entities on shop-floor is described and illustrated by an example. Second, the embedding of the model in the traditional layer structure of the factory layout is show. This chapter ends with Section 5.3 in which the third core element – the *quality-oriented workflow* – is described. The illustration of the approach for a quality-oriented workflow starts with the general scope of the approach. After that, the orchestration of the production by the Workflow Managers, the management of the quality by the Quality Managers and the capabilities for dynamic routing are shown. In each section, aspects for bringing the proposed approaches into industrial practice and an illustrating example are provided. Each section ends with a discussion on the novel aspects and potential drawbacks of the respective approach. Furthermore, a validation of the approaches against the requirements identified in Section 3.6 is done.

5.1 Task-driven manufacturing

In this section the most important element of the architecture for flexible manufacturing is described. The reader is guided step-by-step through the approach of task-driven manufacturing.

5.1.1 The principles of task-driven manufacturing

The flexible behavior in process execution on shop-floor is obtained by Manufactrons which are encapsulating the respective knowledge in the machines and devices. Depending on the task to be fulfilled and the desired output, the Manufactron is supposed to decide by itself on how the process shall be performed. The input information for a Manufactron is composed only by such information which *describes* the task. It is in the responsibility of the Manufactron to decide on *how* the task will be performed. The general sequence of task-driven manufacturing is illustrated in Fig. 31:

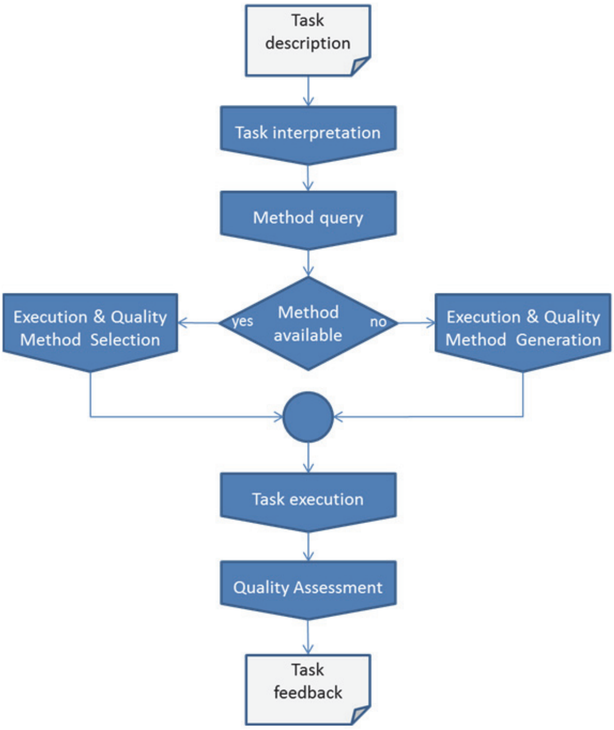


Fig. 31. General sequence of task-driven manufacturing.

The sequence starts with the receiving of a task description in which the task itself and its boundary conditions are described. Next, the Manufactron queries its knowledge system in order to identify a proper method which is capable to fulfill the task. If the task is not available in the knowledge system, an (iterative) task generation process is started. After that, the task is executed and an assessment of the quality of the task execution result is done. Finally, a feedback on the task execution quality is issued.

Tasks are processed synchronously, meaning that a new task can only be processed when the previous task has been finalized completely. However, the task processing sequence can be interrupted by internal and external events. In any case, a task feedback is issued in order to guarantee a complete documentation of each processing step.

5.1.2 Task-driven operation sequence

In this section, the different building blocks of the processing sequence for task-driven manufacturing are illustrate in more detail.

Task interpretation

The first activity after the Manufactron received the task description is to interpret the task. The task interpretation basically has two main goals. The first goal is to check if the described task fits the skills of the Manufactron. At the beginning the task description is checked on semantic correctness and consistency of the given information. In case of incorrect or incomplete information, it is the responsibility of the Manufactron to continue with the operation of the task by a task reconstitution or to reject the task. If all information is correct, the Manufactron verifies if it is principally able to fulfill the task by matching the task description with its own capabilities. As the task description is tailored to the Manufactron type and is addressed explicitly to a dedicated Manufactron within the production (no broadcasting is done) in the normal cases the Manufactron is capable to fulfill the job. The second main goal is to extract the features which are describing the task and the relevant boundary conditions. Both data is used by the subsequent method query component for querying the task execution and quality assessment methods in the Manufactron's knowledge system.

Task to method mapping

After the task has been interpreted, the next step is to find a proper method for the task execution. The mapping of an executable method to the task description is the core functionality of the Manufactron. The task to method mapping is a part of the Manufactron's knowledge system which encapsulates the knowhow on the respective production process. It builds the "intelligence" of the Manufactron and also reflects the process knowhow of the vendor of the Manufactron. The inputs of the task to method mapping process are the features which have been extracted from the task description during the task interpretation process. Fig. 32 illustrates the three major steps of the task to method mapping process of a Manufactron.

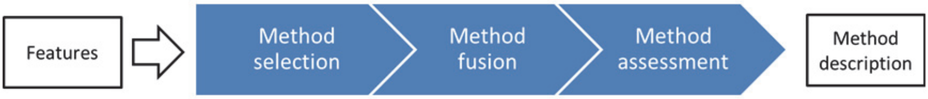


Fig. 32. General scheme of the mapping of task descriptions to methods.

The first step of the task to method mapping is to select proper methods for process execution. The selection algorithm highly depends on the way the process is modeled. For that several approaches do exist, for example:

- Databases in which the methods for fulfilling the tasks and of quality assessment methods are stored. As companies already use databases for storing and maintaining information on processes, machines and products, this approach helps for a fast take-up of the architecture;
- Decision trees or tables which consist of rules for the automatic mapping of task features to methods. This approach is useful in such cases where process knowhow is elicited from human experts as engineers, maintenance people, etc. are often able to describe process behavior in “if… then…else” rules;
- Functional mapping of tasks to methods.

If required, a combination of the technologies can be used. The result of the method selection is a set of the best fitting methods for the task. In the next step, a fusion of the methods is done. The set of methods is processed by assessing the individual results of each method. At the end, the method which is most appropriate for the task is available. The final step of the task to method mapping is the validation and assessment of the method. This is done by calculating the confidence value for the method. The confidence value is used by the Manufactron to estimate if the task to method mapping was successful or if the task cannot be fulfilled.

The functionality and the reliability of the task to method mapping are limiting the Manufactron’s capabilities of being flexible and adaptable on varying manufacturing conditions. Table 7 illustrates the major limiting factors in the task to method mapping.

Table 7. Major limiting factors for successful task to method mapping of a Manufactron.

Factor	Description
Sufficiency and precision of the task description	The better the task description reflects the real working environment of the task execution, the better is the likeliness of finding a proper method.
Range of process methods	The range of process methods reflects the general working space of the Manufactron. The more methods are available and the wider the range of the methods is, the higher is the likeliness of finding a proper method for a given task description.
Density of process methods	The density of process methods reflects the number of methods within a certain range. The higher the density, the higher is the likeliness of finding a proper method.
Quality of process methods	Only if the available process methods are reliable and their execution leads to the desired result is the Manufactron able to perform properly. This is not a special condition for the task-driven operation, but a general requirement for process execution.
Quality of features for method identification	The features build the basis for the method query. Only if these features describe the process properly can a reliable query result be expected.
Quality of the method identification algorithm	The method identification algorithm is the way for finding a proper method in the Manufactron's knowledge system on the basis of the features extracted from the task description. Proper algorithms must be available in order to guarantee reliable method identification.
Quality of method fusion	The quality of fusion algorithms for the set of methods which are returned as a result of a knowledgebase query. This targets capabilities e.g. for data interpolation and extrapolation.

Task execution

After a proper method is available, the method is executed. In case of a production task this means to download the method to the respective device which physically performs the execution of the process. Examples of such devices are robots, Programmable Logic Controllers (PLC's) and Numerical Controls (NC's). The physical execution in the task-driven manufacturing paradigm is analogous of any other process execution. Naturally, the presentation of the method and the communication scheme between Manufactron and physical device depends on the capabilities and interface specification of the device. For that reason, both are vendor and/or device specific. In order to take this into account, the Manufactron is capable to generate device-specific methods out of the more generic and device-independent knowledge system.

Task quality assessment

The final step of a task-driven operation sequence is the assessment of the task result after the physical execution on the device. The way of processing the quality assessment is very similar to the processing of process methods. The methods for quality assessment are also a part of the Manufactron's knowledge system. Depending on the task, proper quality assessment methods are selected. After the method fusion and validation, the result of the process execution can be assessed. To do so, the physical process is monitored and relevant process factors such as process signals, events and messages are recorded. Based on this, relevant features are extracted which are subsequently used as an input for the result assessment. The methods for task assessment can be implemented in a generic manner by using approaches such as data mining, neural networks, classifiers, etc. However, the input information needs to be adapted according to the physical process data. After the assessment of the task result, a task feedback is compiled and issued to the quality managing entities on MES level. The detailed mechanisms for quality managing are discussed in Section 5.3.2.

5.1.3 Task-driven devices

After the reader now is familiar with the mechanisms of task-driven execution of manufacturing jobs, this section describes the structure of Manufactrons. After a general description, the internal components and the interfaces of such devices are illustrated.

General description

Traditional equipment such as ordinary controls or devices is not able to perform task-driven jobs. Some functionality which is required for task-driven processing such as capabilities for self-description and process emulation is already partially integrated in commercially available equipment. However, in order to realize task-driven manufacturing, devices have to be composed of several mandatory internal building blocks and interfaces. Manufactrons are designed to be compatible to traditional equipment. Traditional equipment is used as a basis and is *extended* by the mandatory components for task-driven processing. The component diagram of a Manufactron is shown in Fig. 33.

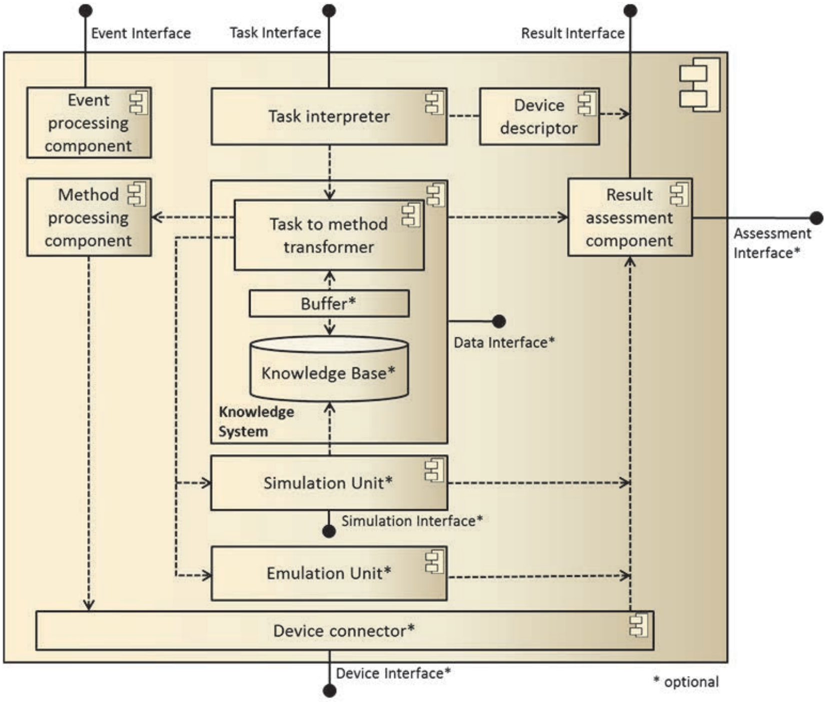


Fig. 33. Component diagram of a Manufactron. Optional components are marked with an asterisk.

A Manufactron is composed of twelve components from which five are optional. The dotted lines indicate the data flow and its direction between the components. Furthermore, a Manufactron has six interfaces to external components from which three are optional. The following two sections describe the components and interfaces in more detail.

Components of task-driven device

This section describes the internal components and building blocks of Manufactron. For each component the purpose and its relation to other components is shown.

The *Task interpreter* receives the task description and interprets the included information. It performs the task check for verifying whether the task description is consistent and if the task is in principal executable. Therefore it is connected to the *Device descriptor*. Furthermore, it extracts the relevant features for the method search. The features are forwarded to the *Knowledge System*.

The *Device descriptor* is responsible for gathering internal data such as states, messages, etc. which is provided to other entities in the manufacturing system or for user feedback. Furthermore, it describes the capabilities of the device for task processing and quality assessment. This information is used in the planning level for the generation of the related task description and quality assessment documents.

The *Knowledge System* handles all the expert knowledge and knowhow on the respective process. To do so, it consists of three subcomponents. The *Task to method transformer* is responsible to find proper methods for the given task description. The method description as the result of the transformation is generic and not tailored to the demands of the control or device underneath. This increases the exchangeability of the devices, e.g. if systems from different vendors shall be connected. The Knowledge System provides both, methods for process execution as well as for task assessment after execution. If required, it uses the optional *Knowledge base* which contains pre-defined methods for task execution. A *Buffer* can be used for pre-loading of methods. This minimizes the time for querying and loading methods from the knowledge base and guarantees the availability of methods under real-time conditions.

The *Method processing component* is responsible for the preparation of process methods for the execution on the respective control or device. It takes the generic method description provided by the knowledge system and converts it into a form which can be understood by the device underneath.

The *Device connector* implements the interface to the (traditional) physical device or control. The first major functionality is to send the method provided by the Method processing component to the connected device. Secondly, it is responsible for gathering process data such as process signals, quality information, etc. from the device. In addition to that, the device connector implements several maintenance functionalities for device handling such as device re-connection in case of device breakdown, alarm and message handling, etc. The Device connector is an optional component. Manufactrons which are implementing only simulation jobs or are used for the coordination of other Manufactrons (see Section 5.2.1) do usually not contain a device connector.

The *Result assessment component* is responsible for assessing the execution of the task. The method for the assessment is provided by the Knowledge System. The input information for the assessment is provided by the respective device connected or by external sensors. The Result assessment component is mandatory for each Manufactron as the issuing of a result description after each task execution is required.

The *Simulation Unit* generates new methods for the Manufactron's Knowledge System. Depending on the individual implementation of the Manufactron, the simulation is triggered in different ways. The first option is to explicitly advice the Manufactron to perform a simulation job. This is done by sending a respective task description and task type (see Section 5.2.2). The second option is to simulate new methods on the fly if the knowledge system does not contain a proper method during task execution. This is only applicable if reliable and fast simulation algorithms are available. The Simulation unit is an optional component because a) simulation is often not possible in real-time and thus, dividing of the real-time and simulation part of a Manufactron in two separate entities might be useful, b) simulation algorithms are not available for all processes, and c) due to legal reasons simulation tools and Manufactron are provided by different vendors.

The *Emulation Unit* performs an emulation of the respective process. In contrast to a simulation, the emulation does not generate new methods. Emulation uses an already existing method of the Manufactron and executes it virtually. The Manufactron receives the task description, queries its knowledge system for a method and delivers the requested key performance indicators (KPI's) to the requesting entity. The emulation mode is usually used in advance of production for finding the optimal production setups, in parallel of production when a rearrangement of the production setup is required or for production optimization e.g. in case of an identified bottleneck. For example, if a Welding Manufactron emulates a task, the result can be the time for executing this task or the energy needed for executing the task without executing the task itself.

The *Event processing component* is responsible for the entire event management. It issues events such as status information to the other entities of the manufacturing system. Furthermore, is handles and distributes events coming from the environment to the internal components. Such events are used for asynchronous communication for example if the task processing should be interrupted.

Interfaces of task-driven devices

For the communication with other manufacturing entities, the Manufactrons are equipped with several interfaces (see Fig. 33). In this section an overview of those interfaces is provided.

The *Task interface* realizes the exchange of task descriptions. It receives incoming tasks and forwards them to the task interpreter for further processing. Details on the implementation of the documents for task description transport are descripted in Section 5.2.2.

The *Result interface* is responsible for issuing the documents carrying the results of the task execution. The result interface receives its data from the result assessment component after the assessment has been done. In case of a request for device description (see Section 5.2.4), the related document are also being issued via the result interface.

In opposite to the two main interfaces for task receiving and result issuing, the *Event interface* implements an asynchronous communication mechanism. Events from and to the device can be sent and received at any time. This mechanism allows for a more dynamic reaction of the system e.g. if a task execution has to be interrupted or if alarm messages have to be issued during task execution. Events are usually used for message handling and are characterized by small portions of data transfer.

The *Data interface* implements the connection to other data sources which might be required for task processing. The most important sources are the Knowledge Systems of other Manufactrons. The data interface realizes the knowledge exchange between Manufactrons located in the same knowledge domain. Furthermore, external databases which contain additional data for task execution such as CAD databases, parameter databases, etc. can be connected.

External simulation tools can be connected via the *Simulation interface*. As (commercial) simulation tools often implement proprietary interfaces, the simulation interface needs also to be adapted for the special requirements of the respective tool.

The *Assessment interface* provides the connection to external (commercial) process assessment tools. By this, monitoring and assessing the process result in terms of quality, timing, stability, etc. can be done during or after process execution.

The *Device* interface implements the connection to the respective device which is executing the production process. Even if standardized interfaces for device interaction do exist (e.g. OPC), in practice proprietary protocols are often implemented. For that reason, the device interface has to be adapted for the special requirements of the devices underneath.

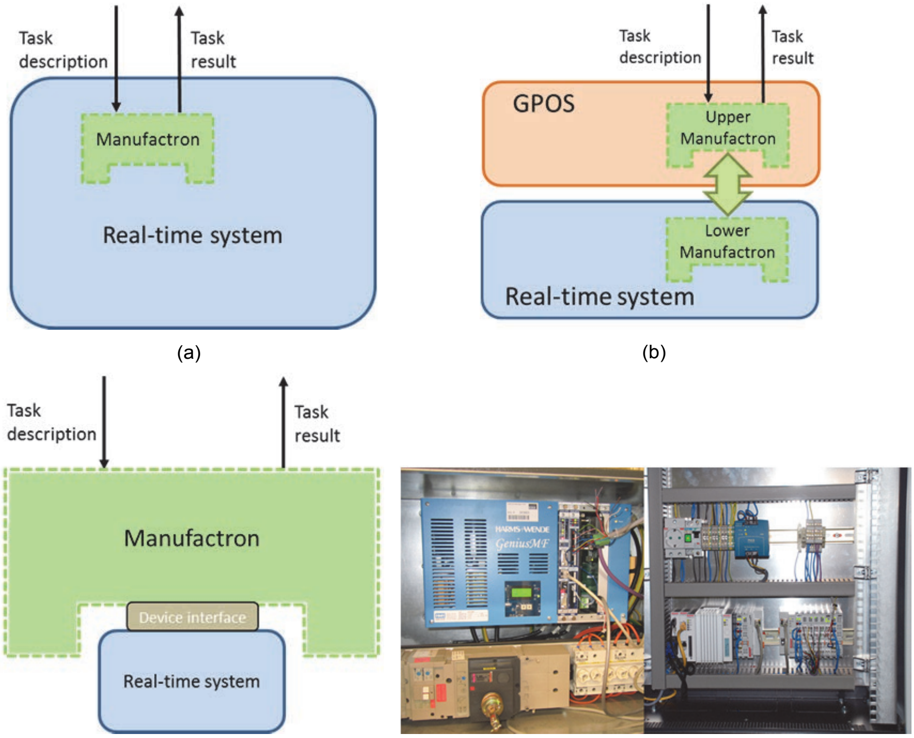
5.1.4 Task-driven manufacturing in practice

As the focus of this thesis is on the design of an architecture which allows for easy implementation in industrial environments, it is also necessary to describe how task-driven manufacturing can be put into practice. As a conclusion of the requirement gathering phase, the most important aspects herein are the interfaces to existing traditional systems within a manufacturing system and the elicitation and formalization of expert knowledge for the Manufactron's knowledge system. For both aspects potential approaches are illustrated and an estimation of the efforts for the implementation of each approach is provided.

Manufactron implementation and interfacing of physical devices and real-time controllers

How to interface existing physical equipment is one of the major questions regarding the implementation of Manufactrons in industrial environments. This question is also linked to the physical implementation of Manufactrons. Therefore, several aspects need to be taken into account. In real-life, environments are often heterogenous, which means machines and components of different vendors or at different places have to cooperate with each other under hard time constraints. The devices and their internal operation sequences are often highly optimized. On the other hand, the implementation of the Manufactron's feature requires additional functionalities such as handling of large, deeply structured data documents, integrated knowledge bases, knowledge-driven decision taking, unified and abstracting layers for data access or the use of semantic technologies.

Against this background, Fig. 34 illustrates three different approaches. In Fig. 34a the Manufactron's functionality is embedded in the real-time operating system (ThreadX, RTLinux, VxWorks, etc.) of the device. Task descriptions and results are communicated via an additional channel. The major advantage of this approach is the possibility of extending the existing device by adding additional functionality to the software only. No additional hardware is required and the connection between Manufactron functionality and the embedding environment is relatively simple. On the other hand, this solution has a number of constraints and restrictions. As most devices are vendor-specific and the software is often tailored to the underlying hardware, a generic implementation of the Manufactron's functionality for different devices is hardly possible and its customization on different devices requires a lot of time and effort.



(a) (b) (c) (d)

Fig. 34. Different approaches for Manufactron implementation and interfacing of real-time systems (Peschl 2008a): (a) Embedding of Manufactron functionality into the real-time system. (b) Splitting Manufactron functionality to real-time system and additional hardware running a GPOS. (c) Implementation of Manufactron in additional hardware running a GPOS and interfacing the real-time system. (d) Implementation of approach (c) as part of the XPRESS project using a Harms&Wende welding controller and a Beckhoff IPC.

Furthermore, performance restrictions of the real-time devices would also lead to constraints in Manufactron functionality (e.g. powerful simulations, parsing large task descriptions, internal data storage) and their implementation could cause negative influences in the real-time behavior. Taking all the positive and negative aspects into account, this solution is deemed most suitable when only relatively restricted Manufactron functionality in comparison to the real-time functionality for process execution is required.

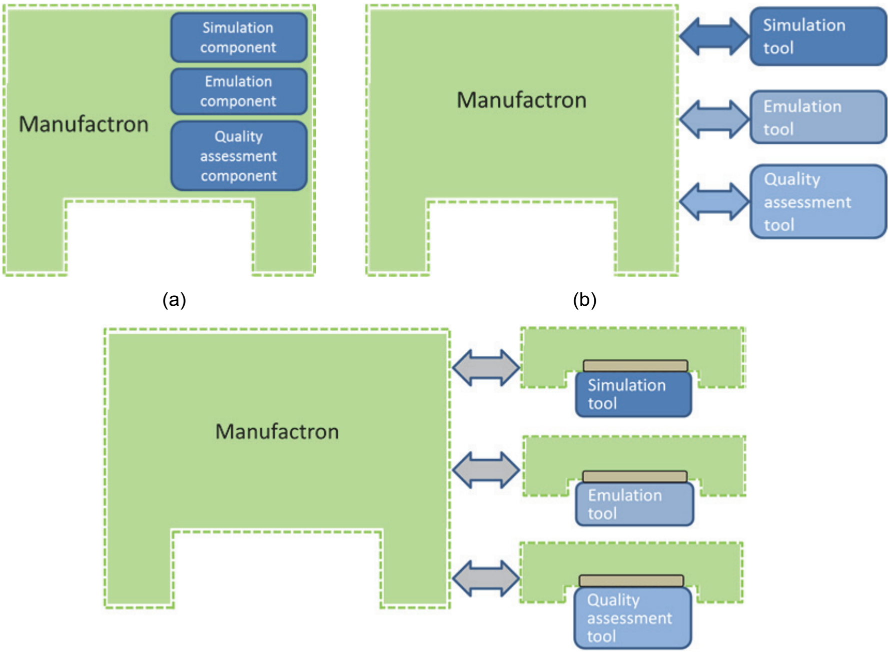
Fig. 34b shows an approach in which the Manufactron's functionality is split. The functionalities which are more related to the process reside on the real-time system while the other functionalities are implemented in dedicated hardware running a General Purpose Operating System (GPOS). This approach allows constraints identified within the former solution to be reduced as the sophisticated functionalities can be implemented in a more powerful environment. In addition, a more generic way of implementing the Manufactron block in the GPOS is possible as long as it is implemented within the same environments (operating system and programming language). However, this solution is more complex as proper communication between the two blocks of Manufactron functionality needs to be implemented. Furthermore, additional hardware is required which adds complexity and costs in terms of investments, maintenance, etc. to the manufacturing system.

In the third solution, the Manufactron functionality and the real-time system are completely separated (Fig. 34c). This way, the Manufactrons functionality can be implemented in a powerful environment, while no adaption of the real-time system is required. It allows engineers to add additional functionality to existing devices without influencing their behavior. This also enables a generic implementation of the Manufactron as all functionality is entirely implemented within the GPOS. Adaptations of the generic structure of the Manufactron are only required within the device connector (also see Fig. 33). The price to be paid in comparison to today's conventional approach is the increasing complexity due to the separated functionality and the additional costs for additional hardware.

Approach (c) was also chosen for the implementation of the Manufactrons in the XPRESS project. Existing devices such as welding controllers, robots of two vendors (KUKA and COMAU) as well as riveting and positioning tools were interfaced by the respective project partners using a *Manufactronic Framework*. The framework was implemented for a Windows environment using the C# programming language and the Windows Presentation Foundation (WPF) by the software companies as part of XPRESS and under the leadership of Fraunhofer IPA, one of the project partners.

Interfacing software tools and data sources

In order to enable the introduction of the task-driven approach to the industry, also existing data sources and software tools need to be interfaced. With respect to the Manufactrons this is related to the Simulation Unit, the Emulation Unit and the Result Assessment Component (see Fig. 33). Simulation and Emulation Unit are designed to simulate or emulate the respective process of the Manufactron such as handling, riveting or welding. This is done in advance or in parallel of production, where no real-time behavior is required. Examples of commercial simulation and emulation tools are SORPAS® for resistance welding processes (Swantec 2013) and RobCad for robot path simulation (Siemens 2009). The Result Assessment Component assesses the result of the process execution. Depending on the respective process, the results need to be assessed after each step and thus, might require fast calculation and reaction times.



(a) (b)

c)

Fig. 35. Different solutions for interfacing commercial software tools: (a) Implementation of functionality within the Manufactron; (b) Direct communication between Manufactron and software tool; (c) Indirect communication by wrapping the software tools in a Manufactron shell.

For quality assessment there are also various commercial products which are tailored to the respective process. Examples of such solutions are HPP-25 for process controlling and data logging for riveting processes (Baltec 2013) or the PQS system for quality analysis of several welding processes such as arc, laser and resistance spot welding (QS-Technologies 2013). Fig. 35 illustrates different approaches for using (commercial) software tools for process simulation, emulation and process quality assessment. The first approach is to implement the respective functionalities as an intrinsic component of a Manufactron Fig. 35a). The effort for the implementation of this solution is rather high as the full functionality needs to be implemented from scratch. This solution is most useful if no appropriate tool is available or when other restrictions hamper an easy tool interfacing (no proper interface available, time constraints, licence related issues, etc.). If these restrictions do not exit, a direct connection to the external software tool is useful. Fig. 35b) illustrates the approach. The communication is provided via the tool interfaces which might be file-based, web-based or via the tool-specific API. The efforts for implementation in comparison to approach (a) are relatively low. However, low reaction times might be taken into account, especially in case of real-time requirements for the quality assessment during or directly after process execution. As part of the XPRESS project this solution was specified and implemented by the author of this thesis for the communication between the Welding Manufactron and the SORPAS® welding simulation tool. Fig. 35c) shows an approach in which the respective software tools are wrapped by a Manufactron shell (see also Peschl *et al.* 2012a). The major advantage of this approach is the standardized communication based on the task description and task result mechanism between the Manufactrons. This enables users to interface software tools in the same way as all other Manufactrons in the manufacturing system. Furthermore, expert knowledge on setting up and using the software tool can be stored in the Simulation Manufactron's knowledge base. By means of this feature the functionality of complex tools can be abstracted and simplified. On the other hand, this solution involves implementation efforts and has also disadvantages in terms of timing. As part of XPRESS this solution was chosen by a partner specializing in factory design for interfacing a complex software tool for line simulation.

Knowledge elicitation and formalization

In order to implement task-driven manufacturing, knowledge on the respective process must be available in the Manufactron's knowledge system. To this end, the first requirement is that knowledge can be identified, meaning that the interdependencies between a task description, the desired task goal and one or more process method(s) are generally known. Furthermore, the task description needs to be complete, meaning that all task variables which influence the result of the respective process can be identified. For proper handling, a generalization of the interdependencies is useful which allows drawing general conclusions on the influence of changing task description variables related to the task goal and the modification of method parameters. As part of the XPRESS project, a portable system for generalized data representation and formalization based on support vector methods was developed which can be adapted to various processes (Pollak *et al.* 2011). The system was tested for data representation and formalization of resistance spot welding processes. Furthermore, a software application called *task-to-method transformation system* was developed which allows for the adaptation to different applications by dynamic scripting (Tuovinen *et al.* 2010). This system was demonstrated for task descriptions and method finding for manual and robotic handling processes. In other projects, for example, knowledge representations based on ontologies (Angelsmark *et al.* 2007) or cloud-based approaches (Skill-Pro 2013) were investigated. The investigations within the XPRESS project showed that the elicitation and representation of knowledge is a challenging task. The more specific process knowhow is required, the more complex is its representation and the task transformation. While for example, handling tasks as demonstrated in Faure (2012) were relatively simple to represent, task transformations in which deep process knowledge is needed more elaborated mechanisms are required, as demonstrated in Eickhorst&Trostmann (2012), for example. In addition, the transferability of methods for task execution also depends on the respective context. Even in traditional approaches, recipes need to be adapted to the special conditions in which they are used. For example, the circumstances and the environment at different factories are different, even if similar production tasks are executed and the origin, which leads to different process behavior, remains hidden. In consequence this implies that the tasks cannot (always) be described in their entire scope. This issue affects the task-driven approach as well as - under certain circumstances - tasks cannot be described in their entirety and only a restricted task to method transformation can be applied. The task-driven approach consists of additional degrees of freedom for task execution in comparison to the recipe-based one. As a result, additional uncertainties are added to process execution. In order to increase the acceptance of task-driven systems in the industry, and especially in the traditional-oriented ones, reliable and easy-to-use quality monitoring and assessment systems need to be available.

5.1.5 Example: Task-driven welding job

In order to illustrate the functionality of task-driven manufacturing, this section provides an example of a task-driven welding job execution.

Whenever metal parts have to be connected, resistance spot welding is one of the most used technologies in industry as it has various advantages in comparison to other joining technologies (see also Rukki 2009). The physical principle of resistance spot welding is to cause current to flow through electrode tips and the separate pieces of metal to be joined. The electrical resistance of the base metal causes localized heating and the weld is made (Miller 2012). Depending on the material types, the thicknesses as well as on various boundary conditions such as material surface condition, the welding parameters have to be adjusted. The main adjustable parameters are the welding current level, the welding time, the pressing force of the electrodes (Rukki 2009). During production, the electrode geometry and electrode material are usually not changed from weld to weld. For that reason they are considered to be constant for the further description of the example.

Fig. 36 illustrates a welding application. The left and the right side of a tin car have to be connected. This is done by welding three welding spots on the roof of the car (marked with red circles).

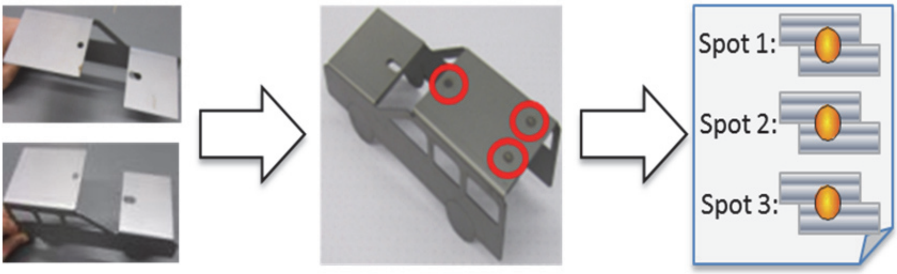


Fig. 36. Example of a welding job with two metal sheets joined by three welding spots.

The welding task is described by the description of the material to be welded. Most important is the type and thickness of materials at each of the three regions where the spots are located (see also Pollak *et al.* 2011). Other parameters which do also influence the quality of the resulting welding spot are:

- The coating of the materials at those regions;
- The surface conditions at these regions;
- The distance between the welding spots;
- The distance of the welding spot to the edge of the metal sheet.

In addition to that, the optimization goal for the welding task has to be described. For the welding process this is usually the quality of the resulting welding spot, defined by the diameter of the spot. Fig. 37 provides an example of a Task Description Document for a welding job:

```xml
<TaskDescriptionDocument >
        <!--
        Definition of the task type, the task objective and the local
        optimization goal
        -->
        <task type="execute" objective="row-spot-welding">
            <property objectId="WeldingSpot0001" property="minWeldingQualitySpot" value="5" />
            <property objectId="WeldingSpot0002" property="minWeldingQualitySpot" value="4.5" />
            <property objectId="WeldingSpot0003" property="minWeldingQualitySpot" value="4.5" />
        </task>

        <!--
        Definition of the task by the description of the
        material types and thicknesses
        -->
        <object id="WeldingSpot0001" name="first welding spot">
            <property property="MaterialType1" value="ST07"/>
            <property property="MaterialType2" value="ST07"/>
            <property property="MaterialThickness1" value="0.8"/>
            <property property="MaterialThickness2" value="1.0"/>
        </object>
        <object id="WeldingSpot0002" name="second welding spot">
            <property property="MaterialType1" value="ST07"/>
            <property property="MaterialType2" value="ST07"/>
            <property property="MaterialThickness1" value="0.8"/>
            <property property="MaterialThickness2" value="1.0"/>
        </object>
        <object id="WeldingSpot0003" name="third welding spot">
            <property property="MaterialType1" value="ST07"/>
            <property property="MaterialType2" value="ST07"/>
            <property property="MaterialThickness1" value="0.8"/>
            <property property="MaterialThickness2" value="1.0"/>
        </object>
</TaskDescriptionDocument >
```

Fig. 37. Example of a Task Description Document for a welding job. Only the main parameters material type and thickness as well as the minimum spot diameter are described. The xml structure contains of tasks, objects and their properties (Hoffmeister *et al.* 2011).

After the Welding Manufactron has received the task description it starts to interpret the task and to extract the features for the knowledge system query. In this example the features correspond to the description of the materials and the local optimization goal. The result of the knowledge system query is a description of a method, both for process execution and for the assessment of the welding quality. The process execution method describes the settings for the main welding parameters current level, welding time and the electrode force. The quality assessment method is usually a software component which allows for monitoring and assessing the welding quality based on measured welding process signals such as electrical resistance, electrode displacement or regulation stroke.

For process execution, the welding parameters are downloaded to the welding controller which physically executes the welding job. In parallel to the process execution, the process data is gathered by either the welding controller itself or by external sensors. After assessing the welding quality, a result document is issued to the upper entities in the manufacturing system. This completes the execution of a welding task. Fig. 38 illustrates the task-driven execution of a welding job.

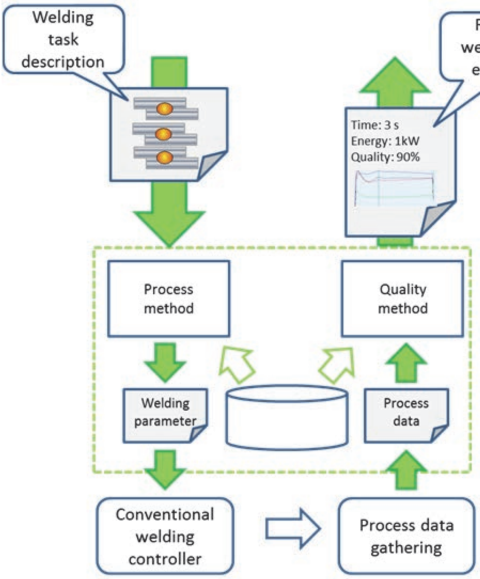


Fig. 38. Task-driven welding job of a Welding Manufactron. The concept and major parts of the implementation has been done by the author of this thesis. The Welding Manufactron has been integrated in the Demonstrators for automotive and electrical industry as illustrated in Section 6.1.2 and 6.1.3.

5.1.6 Discussion

The approach for task-driven manufacturing systems is analyzed and discussed in the final sub section of this chapter. The first part discusses, the potential improvements regarding the implementation of the proposed approach and the constraints with respect to the blocking points for a wider adoption of agent-based manufacturing systems identified in Section 2.5. After that, the several requirements consolidated in Section 3.6 which are related to this topic are reflected and the contribution of the task-driven approach to the fulfillment of those requirements is provided.

With respect to the blocking points of current scientific approaches, positive and negative aspects of the task-driven approach can be identified. On the positive side, the mostly poor interfacing of physical devices by current scientific approaches as identified by Leitão (2009) is taken into account in the proposed architecture. Even though the core of the Manufactrons can be considered as a pure software solution, the interoperation with the related physical (real-time) system is taken into account. Several solutions for interfacing physical devices or even the embedding of Manufactrons into such devices have been illustrated. It is up to the device vendors to decide which solution is finally being implemented. They need to take all major aspects into account and are probably forced to provide different solutions depending on their customers' needs. Embedding Manufactrons into existing real-time systems is probably the most effective solution, but requires modifications of those systems. This makes upgrading more complex and can have a negative impact on the real-time behavior of the devices. On the other hand, the implementation of the Manufactron outside of the real-time systems requires a dedicated device which is more cost intensive and also adds complexity to the overall system. The positive aspect of this solution is the better interfacing of different heterogeneous devices, enabling a more generic implementation of the Manufactron.

The Manufacton's knowledge-based approach also involves a positive and a negative aspect. On the positive side, the task-driven approach contributes to one of the major topics of the research agenda of the Manufuture initiative (MANUFUTURE 2004). Today's programming of physical devices can (partially) be replaced by the Manufactron's knowledge system and the implementation of respective task to method transformations. This allows device vendors to exploit their specific process knowledge which generates additional value. On the other hand, such business models are in contradiction to a major blocking point. As end users usually hesitate to shift the responsibility of component development and maintenance to the respective component suppliers (Leitão 2009), the potential for such a change is currently restricted. In addition, this responsibility transfer also implies a closer cooperation of end users, system integrators and equipment suppliers as the equipment suppliers need to know more about the end users manufacturing processes in order to be able to adapt their equipment to the given environment (see also Sundermeyer&Bussmann 2001).

Taking the aspects mentioned above into account, the proposed architecture is able to provide a number of potential improvements for flexible manufacturing systems. Table 8 shows the potential offered by the task-driven approach to fulfill the requirements identified in Section 3.6.

Table 8. Potential offered by the task-driven approach to meet the requirements of flexible manufacturing systems.

Requirement	Contribution
Flexibility in the introduction of new products	Manufactron's allow for distributed process simulation at shop floor level (Peschl *et al.* 2012a). This can support the factory planning process by also taking relevant behavior of the processes such as process execution time or process quality goals into account. This way, the factory simulation can be more detailed and, the adaptations of device configuration during ramp-up time can be reduced.
Reconfigurability as a reaction to changing demands	Sophisticated task to method mapping could provide modified methods according to the new demands. This enables a (restricted) dynamic behavior of the process execution.
Learning ability of components	Manufactrons can be featured with functions for the generation of new methods. This allows for continuous learning and for the improvement of their knowledgebase. Furthermore, Manufactrons monitor their own behavior during the process execution and are therefore able to draw conclusions on changed boundary conditions for the process quality.
Scalability and extensibility of the system	Manufactrons are not restricted to a certain industrial environment. They can be moved from one system to another. Their knowledgebase can be re-used, but needs to be adapted to the local conditions.
Re-use of existing systems	Conventional equipment can be used and extended to a Manufactron by implementing a Manufactronic shell. Several options for interfacing of existing software tools are available.
Deterministic behavior of the system	The behavior of Manufactrons is determined by the task description and the boundaries for process execution can be set to fixed limits, enabling. a "controlled flexibility" in process execution.

Requirement	Contribution
Acceptance of existing system boundaries	The assignment of responsibilities at shop-floor level is not touched by the Manufactronic approach. This limits the risks for the acceptance of the approach in the industrial environment.
Knowledge-based manufacturing at all levels	The knowhow of the Manufactron vendors can be encapsulated in the knowledgebase and in the algorithms for task to method mapping. This enables vendors to create new business models, e.g. selling process methods for dedicated processes or providing services for quality assessment.
Abstraction of specific knowledge	No specific knowledge for the integration of Manufactrons is required. The Manufactrons are interfaced only by the task description which is (usually) vendor independent. Vendor-specific items are described in the Manufactron self-description.
Traceability of product and process quality	Each request is answered by a Manufactron with a respective result. As a result, each Manufactron operation can be traced. Furthermore, Manufactrons can include quality assessment components which allow for quality control.

5.2 Hierarchical organization and communication model

In the previous section the task-driven processing of controls and devices has been described. With this, the basis of the novel architecture is available. In this section, the details of embedding task-driven manufacturing equipment in the factory are illustrated. First, the concepts for hierarchical Manufactron organization on the shop-floor are described. After that the mechanisms for hierarchical communication and for the synchronization of Manufactrons during operation is shown. In addition, the most important aspects for industrial implementation of the proposed hierarchical organization and communication mechanisms are discussed. Next, an example of the collaboration of two Manufactrons which are performing a combined handling and welding job is illustrated. This section ends with a discussion on potential improvements and blocking points identified in Section 2.5 and with a reflection of the requirements identified in Section 3.6.

5.2.1 Machine and system organization on the shop-floor

Manufacturing tasks are usually performed by a number of different controls and devices. In Section 4.2.2 the general architecture for the orchestration of manufacturing equipment has been illustrated. Following this approach, the devices and controls on the shop-floor are organized in a modified hierarchical architecture. The details of this hierarchy type have been discussed in Section 2.1. The control components which are responsible for the coordination of the underlying sub-components, are defined as *Super Manufactrons* in the task-driven architecture. Super Manufactrons are supposed to coordinate the underlying Manufactrons in order to perform more complex manufacturing tasks. For that, several coordination mechanisms such as distributed state machines as described later on in Section 5.2.5 are available. The underlying Manufactrons which are coordinated by a Super Manufactron are defined as *Sub-Manufactrons*. The general organization scheme is illustrated in Fig. 39.

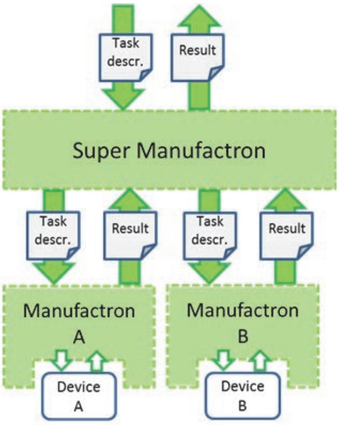


Fig. 39. General organization scheme of a Super Manufactron and underlying Sub-Manufactrons.

The general working scheme and the internal structure of Super Manufactrons is similar to ordinary Manufactrons. Super Manufactrons can be connected to physical devices or as illustrated in Fig. 39 act as pure software entities if they have a coordination role only. Similar to other Manufactrons, Super Manufactrons receive a description of the task to be performed and transform the task description to a method. However, tasks for Super Manufactrons do not describe the goals of a single process execution, but the overall goal of the domain for which the Super Manufactron is responsible for. There are no specific rules on the segmentation and building of those domains and how the hierarchy needs to be derived. In the frame of the XPRESS project Super Manufactrons came into play when an overall coordination of various specialized Manufactrons was required in order to reach a superordinate production goal. If for example, a Super Manufactron is responsible for the coordination of a manufacturing cell in which a car door is assembled, the Super Manufactron receives the task description "*Make car door*" and the related boundary conditions. It is in the responsibility of the "*Car Door Super Manufactron*" to decide on the best way to perform the job. Again, this is stored as a method in the Super Manufactron's knowledge system. In this case, the method describes the knowledge on how to assemble the car door under the given boundary conditions, e.g. which equipment is required, which task descriptions have to be issued for the underlying Sub-Manufactrons and how the Sub-Manufactrons are synchronized. At the end of the task execution, the result of the task performance is issued. To do so, the Super Manufactron gathers and assesses all task result information of the underlying Sub-Manufactrons and compiles an overall result document. The method for quality assessment is again related to the Super Manufactron's specific knowledge and therefore an integral part of its knowledge system. In principal it describes how to assess to overall quality of the job performed by the assessment of the results of the single underlying Sub-Manufactrons and the combination of those assessment results.

In case of manufacturing setups which are not too complex, the building of a two level hierarchy as shown in Fig. 39 is sufficient enough. However, for more complex setups, additional hierarchical levels might be useful. For that reason, the Super Manufactrons allow the building of multi-level hierarchies on shop-floor. This helps to reduce the complexity of large system setups. On the other hand it also increases the communication intensity between the different Manufactrons and the efforts of designing and maintaining proper task descriptions.

Fig. 40 shows an example of a four level Manufactronic hierarchy and the contents of the respective task descriptions. The Super Manufactrons of each level abstract the orchestration of production entities or a certain production steps respectively. The machines and controls are orchestrated by the Manufactrons on Level 1. In the typical cases only the Manufactrons on the first abstraction layer are connected to real-time systems. The content of the task descriptions for those Manufactrons consist of the general functionality of the production equipment underneath. On Level 2 the Super Manufactrons come into play. They are representing more complex structures such as cells or line sections and orchestrate the Manufactrons of Level 1. The content of the respective task descriptions reflect the task goal of the respective manufacturing structure, for example "*make sub-assembly*". If required, further hierarchy levels can be introduced.

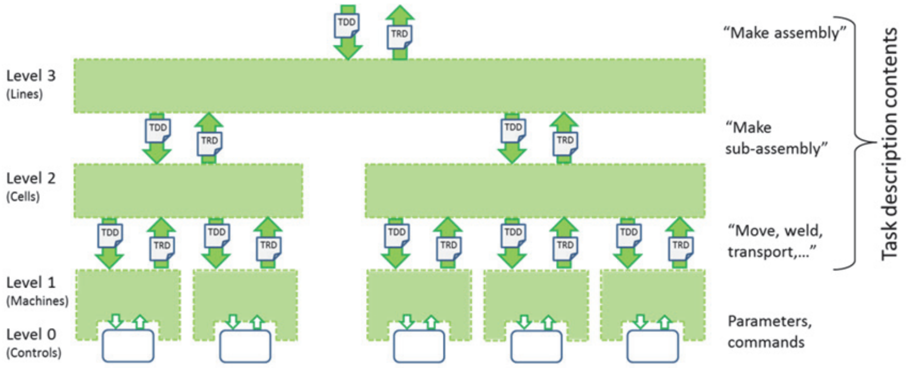


Fig. 40. Example of a multi-level hierarchy of Super Manufactrons for building complex manufacturing setups.

The introduction of the Super Manufactrons provides a solution for the limited knowledge exchange capabilities identified in Section 4.6. Super Manufactrons allow for knowledge domain bridging of the underlying Manufactrons. Knowledge of different domains is interlinked in order to perform manufacturing tasks for which expertise of different domains is required.

Opposite to other approaches using coordination units for hierarchical control (e.g. Leitão 2004), the grade of flexibility in the assignment of the underlying Manufactrons to the Super Manufactron is more restricted. The assignment of Sub-Manufactrons to a Super Manufactron and the related communication and synchronization protocols are defined during the setup of the manufacturing system using the emulation and simulation capabilities of the Manufactrons. During the execution phase the setup of the Super Manufactron's domain is not changed anymore.

5.2.2 Task Description Documents

Task Description Documents (TDD's) are supposed to transport relevant information on the description of the task within the manufacturing system. The content of the Task Description Documents specifies what a production entity has to do in which way and specifies boundary conditions for the task performance. The information is structured in a standardized form within the Task Description Documents. The following two sections describe the content in more detail.

Task definition within the Task Description Documents

The Task Description Document is the only source of information for a Manufactron for its task performance. Or in other words, the Task Description Document consists of all data which is required by a Manufactron for performing the desired production step(s). The basic elements of a production step are the definition of the objective for the production step, the final output after performing the task and some context information. For that reason, Task Description Documents consist of three main elements:

Task objective. The task objective describes *what* shall be done in order to fulfill a certain production step. It usually describes a process to be triggered, e.g. "spot welding", "riveting", "gripping" or "moving". Simple controls and devices are only able to fulfill one specific task objective. In this case, this information might be redundant and will only be used for a plausibility check during the task interpretation. More advanced controls are also capable to fulfill more than one task objective. For example, welding control units are available which are capable for resistance spot welding as well as for seam welding. In those cases the task objective specifies which of the alternatives will be used.

Task goal. The task goal describes a situation after performing the task. It is usually the information on the desired output of the production step. In case of a handling task to be performed by a robot, the task goal might be the description of the object to be moved and the end position of the movement including the information of the desired precision.

Task boundary conditions. Boundary conditions describe the context in which the task shall be fulfilled. The boundary conditions provide additional information on the task performance. Often, they define restrictions or describe more precise situations on the task performance.

Task objective, the task goal and the task boundary conditions are in the relation "*Perform the task objective in order to reach the task goal in the context of the task boundary conditions*". Having this information, the basic information of a production step is available and the Manufactron is able to perform the task.

By the definition of the task objective, the task goal and the relevant task boundary conditions, the Task Description Document consists of all data which is required by a Manufactron. As described above, the task description is the only source of information and no other interfaces do exist. By keeping the interface to the Manufactron small and generic, maximum exchangeability is guaranteed. In order to support also demands on data embedding which are specific to the

Manufactron and which are currently not envisaged, Task Description Documents allow specifying links to “external” data sources. Such links are defined as hyperlinks within Task Description Documents and can target any data source which is required to describe the tasks entirely. In practice this feature can also be used to link to huge amounts of data, e.g. CAD data for the description of objects, which embedding would lead to an explosion of the size of the Task Description Documents and make them hard to read for humans. On the other hand, this feature leads to a risk of unavailable data e.g. due to broken links. From practical point of view a good balance of the pros and cons of using external links has to be done during the manufacturing system design.

Task types of a Task Description Document

Manufactrons are capable to distinguish between different operational modes. The task type indicates in which *mode* the task shall be carried out or in other words, what to do with the given task description. The following three modes are available:

Execute. Advises the Manufactron to execute a physical task on the related production device. Execute commands trigger the Manufactron to query a method in its knowledge system and to download the best fitting method found to the production device. The execute command is the most relevant during production. Due to the real-time requirements, execute commands and the related functions (e.g. task parsing, knowledge base query) have to be time optimized.

Simulate. A task shall not be executed physically on a production device, but only be simulated. The result of the simulation is a new method for the task, which can be inserted into the Manufactron’s knowledge system and can further be used for task execution. Simulation tasks are required in such cases, where no (proper) method for a task description is available. Due to time constraints, simulation tasks are usually not triggered during production. Due to long calculation times and a subsequent validation of the simulation results (see Section 4.2.2) simulation tasks are done in advance or in parallel of the production using redundant equipment.

Emulate. In opposite to the „simulation“, the emulation does not generate new methods. The emulation command triggers the Manufactron to perform a virtual execution of an already existing method of its knowledge system. Hence, the Manufactron shall behave without connected hardware the same ways as with connected hardware and real products. Restrictions on this capability might be accepted. The emulation mode is usually used for functional simulations on higher Manufactron levels. For example, if the Welding Manufactron emulates a task, the result can be the time for executing this task or the energy needed for executing the task without execution the physical welding task.

The task type command enables to Manufactron to operate in different modes using the same communication model. By this, a generic interface is available which allows for a smooth integration of all operational modes in one intrinsic communication model. If for example a method is not available for execution, the Manufactron is enabled to decide on the generation of a new method on the fly (see Fig. 31 on page 92). Using the emulation capabilities, the whole manufacturing system including all communication and data flow between the productions entities can be designed virtually in advance or even in parallel to the real production.

Besides the three main task types, three additional modes do exist. They support various functionalities which are commonly required: The *Self-description* task type is related to the intrinsic features and capabilities of a Manufactron. The Manufactron sends a description of its type, behavior, capabilities and other static data. For task-driven manufacturing this feature is mainly required for the generation of the contents of the task description and the related boundary conditions during production ramp-up. With the self-description the Manufactron describes its capabilities and limitations which is used on ERP level to decide if a Manufactron can be used for a certain job and which information it requires for the job fulfillment. With the *Status* task type the Manufactron provides information on its current status. To do so, various KPI's and other relevant data is compiled and issued by the Manufactron. The *Data* task type triggers the device to provide process data and the *Process history* task type requests a history of the performed processes.

Additional elements of Task Description Documents

Beside the three mandatory elements: task description, task type and task boundary conditions, a Task Description Document can also contain additional elements (Hoffmeister *et al.* 2011). Supplementary information with respect to the task description might be the description of *objects* which are involved in the task execution. This is relevant for example in handling tasks. Objects can define the geometry of the components to be handled or the location of obstacles. Several *Situation* statements can be used to describe the required situation before and after task execution. If for example the task is supposed to perform a coating to a metal sheet, "uncoated" can be expressed in an in-Situation is and "coated" in an out-Situation. Situations are most helpful for process chains where the result of a process step influences the performance to the following process step.

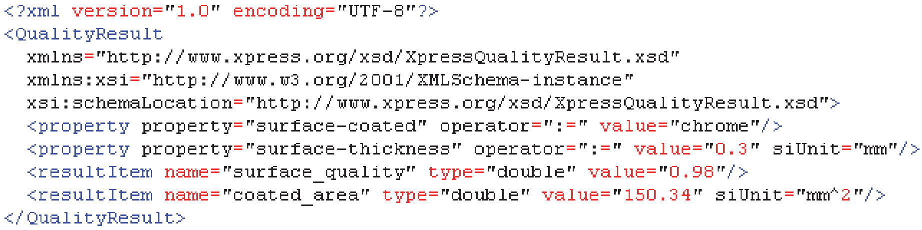
Another optional component of a Task Description Document is the description of requested quality items. By this, a Manufactron can be triggered to issue the requested quality items after the task execution. This feature can be used to select only the relevant quality items from a Manufactron within the given conditions e.g. in order to avoid unnecessary network traffic.

5.2.3 Quality Result Documents

After the processing of a Task Description Document the Manufactron compiles the result of the operation in one document and returns it back to the sender of the Task Description Document. Documents containing the result of an operation are the so-called *Quality Result Documents* (QRD's). The content of the Quality Result Documents principally depend on the task type specified in the Task Description Document. The Manufactron compiles the answer on a task request in relation to the content of the task type. There is no standard or a "must have" of available data the Manufactrons can provide. In fact it depends on the specific capabilities of each single Manufactron which tasks can be performed and which data can be provided within the Quality Result Document. The description on the available data items for the task feedback are – similar to the description of the capabilities of the task performance – defined in the Manufactron self-description (see Section 5.2.4) and can be requested via the task type *self-description*. Depending on this, the requesting entity within the manufacturing system is able to compile a Task Description Document in which the request for the required data items of the Quality Result Document is included.

In case of an emulation and simulation task, the Quality Result Document consists of the relevant KPI's of the simulation/emulation job. The result of a request for the self-description and status of a Manufactron is information on its capabilities and its current configuration. In case of a data and a process history request, the Quality Result Documents might consist of process data such as process signals, state information, etc.

The major application of the Quality Result Documents is to transport the information on the quality of the performed production process. To do so, the Quality Result Documents contain a list of quality data items, which are corresponding to the quality request of the Task Description Document and are reflecting the tasks performed. A quality item includes the name of the quality item, the data type, the unit and the quality value (Hoffmeister 2012). The generated quality data can either be process-specific or none-specific. A process-specific quality value for resistance spot welding is e.g. is the diameter of the welding spot. A none-specific could be a normalized quality value in the range of 0 to 100. With the quality data, also descriptive information such as the unit, the data type, etc. is included. Furthermore, the Quality Result Documents can contain property information. With this, the influence of a task execution on the properties of the objects can be described. For example, the fact may be stated, that the metal surface of the work piece is now coated with a specific material. With this a subsequent production step is able to assess if the properties of the incoming component or material are correct. Fig. 41 provides an example of a Quality Result Document after performing a coating task:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<QualityResult
  xmlns="http://www.xpress.org/xsd/XpressQualityResult.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xpress.org/xsd/XpressQualityResult.xsd">
  <property property="surface-coated" operator=":=" value="chrome"/>
  <property property="surface-thickness" operator=":=" value="0.3" siUnit="mm"/>
  <resultItem name="surface_quality" type="double" value="0.98"/>
  <resultItem name="coated_area" type="double" value="150.34" siUnit="mm^2"/>
</QualityResult>
```

Fig. 41. Example of a Quality Result Document for a coating job.

The example of a Quality Result Document above consists of the two result types after the coating operation. The *property* attributes indicate that the work piece is now coated and provide also information on the coating properties. Furthermore, the *resultItem* attributes describe the quality of the coating in more detail.

5.2.4 Documents for self-description of equipment

The Manufactron self-description describes the specific expertise, needs and limitations of a Manufactron. The documents for self-descriptions allow for discovering and exchanging the properties and capabilities of Manufactrons within the manufacturing system. Such capabilities are especially required during the design phase when the setup of the manufacturing system is determined. Having this, the system designer is able to decide whether a Manufactron is capable to perform a required manufacturing job and which costs are related to it.

Furthermore, the self-description specifies which data items the Manufactron requires in the Task Description Document and which data items are issued by the Manufactron within the Quality Result Document. The latter is required to decide whether the Manufactron is capable to deliver proper quality items after a task operation. The self-description is also needed for the (auto-) generation of Task Description Documents on ERP level as it specifies the tasks, properties and quality items which the Manufactron is capable of. The self-description is restricted to configuration and operating data. The capabilities of a Manufactron referring to its methods are defined by the Task Description Documents which the Manufactron is able to interpret. Table 9 provides an overview of the items of the Manufactron self-description documents and a description of each item.

Table 9. Items for the self-description of Manufactrons.

Item	Description
Description	Static information on the Manufactron in general, e.g. name, serial number, vendor of the Manufactron.
Task capabilities	Description of the task definition and task type the Manufactron can perform.
Pre-properties	The pre-properties which are required in order to fulfill a task goal successfully. The pre-properties are set during production by the previous Manufactrons in the production sequence.
State machine data	Data which is required for the synchronization of Manufactrons by a Super Manufactron (see Section 5.2.5).
Quality items	The quality items the Manufactron is capable to provide via the Quality Result Document.
Connection information	Information on the connections of the Manufactron to other Manufactrons.

5.2.5 Synchronization of equipment during operation

As depicted in Section 5.2.1, a hierarchical structure for the organization of Manufactrons on the shop-floor has been chosen for the architecture's design. Super Manufactrons are supposed to act as the coordinators of several Sub-Manufactrons. This section describes several synchronization mechanisms for Manufactron collaboration during operation.

Synchronization via Task Description Documents and Quality Result Documents

In this approach, the usual Manufactron communication mechanism via the Task Description Documents and the Quality Result Documents is used. All the communication is done via a Super Manufactron which acts as a master of communication. It triggers the start of the collaboration by issuing a Task Description Document to the first recipient (e.g. Manufactron *A*). After performing his job, Manufactron *A* sends a Quality Result Document back to the Super Manufactron. The Super Manufactron issues the next Task Description Document to the Manufactron *B* and waits for the Quality Result Document and so on. The advantage of this approach is its easy implementation. The usual communication mechanism of the Manufactrons is used and no additional efforts are needed for implementation of the collaboration. The main disadvantage of the approach is the limited communication speed. The task description and task result must be parsed every time once one of the collaboration partners receives a respective document. In practice, this could lead to a reduction of the cycle times.

Synchronization via events

In this approach, the event interface of the Manufactrons is used for the synchronization. Events are used to communicate messages asynchronously between the different entities of the manufacturing system. As described above, events usually do only consist of small portions of data which can be transferred and processed rapidly. Depending on the setup of the manufacturing system, the collaboration is either triggered by a Super Manufactron or by the Workflow Execution System (see Section 5.3.1). After this, the cooperation is entirely done by the Manufactrons. The collaboration via events is also a standard mechanism of Manufactron communication. For that the main advantage again is the fast implementation of a collaboration sequence. In addition, the communication speed is quite fast in comparison to the synchronous document exchange. No parsing of messages is required and tasks can be stopped at any time by sending asynchronous interrupt events.

Synchronization via distributed state machines

The most sophisticated approach for the synchronization of Manufactrons is to manage the collaboration via distributed state machines. To do so the Super Manufactron creates a state machine for all sub tasks of the underlying Manufactrons (Hoffmeister 2012). The state machine comprises all the states which have to be executed in the safe or real-time context. The Super Manufactron communicates the state machine as a part of the Task Description Document to all underlying Manufactrons. As soon as the Manufactron receives the Task Description Document it activates the state machine. After this, it sends out the command of being ready for the collaboration to the Super Manufactron via a separate communication channel. The Super Manufactron monitors the incoming signals of each related Manufactron. As soon as all Manufactrons are ready for collaboration, the Super Manufactron initiates the transition to the next state. From now on every transition is synchronized in a similar way. The collaboration ends when the underlying Manufactrons signalizing their final state to the Super Manufactron. The implemented states and transitions as well as the related identifiers are solely the responsibility of the respective Manufactron and is independent of the collaboration as such. The information on the capabilities for collaboration can be requested via the Manufactron self-description. Before the collaboration is initiated, the Super Manufactron requests the self-description of all collaboration partners and compiles the state machine with respect to the specific state names and/or identifiers (Hoffmeister 2012). By this, the underlying Manufactrons do not need to know about the capabilities of the other collaboration partners. This avoids the modification/adaption of the state machine of each collaboration partner to the actual collaboration scheme. In addition to that, equipment of different vendors can be used for collaboration without modifying the Super Manufactron’s collaboration code as long as the Manufactron is in principle able to fulfill the collaboration tasks.

The main advantage of the collaboration via distributed state machines is that only the transitions must be communicated between the Super Manufactron and the underlying Sub-Manufactrons. The amount of data which must be communicated is very small and no time intensive parsing of documents has to be done. Provided that a fast communication channel is available, deterministic collaborations with short processing times can be implemented.

Other synchronization mechanisms

Beside the approached outlined above also other synchronization mechanisms are feasible (e.g. Petri Nets) which have however not been investigated within this thesis. It is finally the decision of the system designer which mechanisms will be used. The relevant decisions criteria are basically the requirements on the collaboration speed and reliability as well as on implementation efforts and costs.

5.2.6 Hierarchical organization and communication in practice

This section provides some examples of how the hierarchical organization and communication mechanisms illustrated above can be implemented in industrial environments. It concentrates on the implementation of the Super Manufactrons and on the communication mechanisms for synchronous communication of Task Description Documents and Quality Result Documents.

Managing and handling of Task Description Documents

As described above the Task Description Documents contain the description of the task which shall be executed by a Manufactron. In this context, a task cannot only be an atomic “command”, but can consist of sub tasks which the Manufactron shall execute in consecutive operation. The architecture does not limit the number of sub tasks within a Task Description Document nor does it limit its size. Task Description Documents are proposed to be implemented in an xml compatible format (Hoffmeister *et al.* 2011). The xml specification targets on an easy-to-use format which is human readable but which can also be interpreted by machines without problem (W3C 2006). This property turned out to be very useful for the definition of Task Description Documents within the XPRESSS project. Manufacturing experts could easily create Task Description Documents using proper software tools without significant training. However, it also turned out that the handling of large Task Description Documents is very complex, especially keeping the internal items consistent turned out to be difficult. Consequently, a manual generation and maintenance of Task Description Documents is only effective within relatively small system setups. For large setups proper tools are required which assist the operators during generation and maintenance of Task Description Documents. Even if such tools are available, the management of large system setups remains challenging. A critical aspect also refers the general structure of xml-based Task Description Documents. Besides the advantages of xml-structured documents mentioned above, there are also a number of disadvantages which play a role in industrial implementation. The major disadvantage is the poor ration between context information and reference data (Murrell 2007, JSON 2011). Therefore, large files need to be communicated via the network and parsed by the Manufactrons. Investigations on the efficiency of xml in comparison to other formats show that xml operation requires significantly more resources in CPU and memory utilization (Nurseitov *et al.* 2009). This issue has also been observed during the implementation of the Manufactrons. Especially the time for parsing the Task Description Documents at the beginning of an operation is an issue as the execution of a task-driven command is expected shortly after the task arrival at the Manufactron.

Task Description Documents in heterogeneous environment

Another important aspect for industrial implementation of task descriptions is their handling and behavior in a heterogeneous environment. The general idea of describing only the task, but not the way how to execute the respective method also targets on a more vendor-independent communication within a heterogeneous manufacturing system. If such a task-driven communication can be established, the usage and exchangeability of devices from different vendors can be increased (see also Section 5.1.3). On the other hand, vendor-specific Task Description Documents require a lot of management effort. A precondition for this is that the task can be described in a generic manner and that those task descriptions can be interpreted by the Manufactrons. Such investigations were not conducted as part of the XPRESS project since no competitive partners were involved. However, some experts coming from the robotic and welding business expressed their doubts on generic task descriptions for two reasons: Firstly, devices coming from different vendors perform process execution in different ways and are based on different assumptions. That is why it might also be required to provide different task descriptions, especially the context information and the level of detail. Secondly, the effective use of generic task descriptions also implies a standardization of the structure and semantics of the Task Description Documents. However, the creation of standards is usually a long-lasting process and lock-in effects of current system implementations can hamper a fast launch.

Implementation of Super Manufactrons

As Super Manufactrons are constructed in a very similar way as other Manufactrons, no significant differences in their implementation exist. In practice they assume the role of the traditional coordination entities in a manufacturing system such as PLCs. In addition to that, they can be implemented as pure software components in order to allow for global optimization, for the coordination of Process Manufactrons and for domain bridging issues. Due to the potential gap between the performance requirements of the Super Manufactron implementation and the available resources, especially in case of PLCs, a dedicated hardware is required to be used for the implementation of the Super Manufactron. This adds complexity to the overall system and leads to increased costs in investment and maintenance. Furthermore, the robustness of the system decreases as Super Manufactrons implement local centralized entities. Finally, the system designers need to balance the benefits and the weaknesses of hierarchical organizations.

5.2.7 Example: Handling and welding production cell

For the illustration of the approaches for hierarchical organization and synchronization, the section provides an example of a real manufacturing scenario. In robot cells as shown in Fig. 42, metallic components are assembled by setting a number of welding spots.

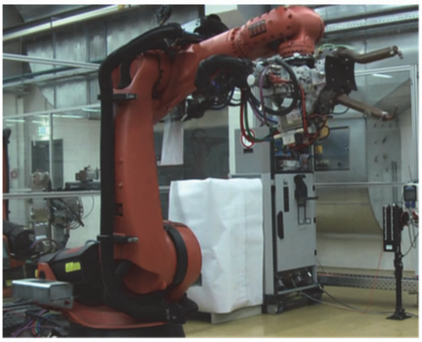


Fig. 42. Typical cell for automotive body shell work: Robot for holding and moving of a welding gun and electrical cabinets with robot controller and welding controller.

For that, the cell is composed of a robot for carrying and positioning of a welding gun and a welding control unit (hidden in the grey cabinet) which is responsible for the control of the electrical power for the welding process. In the traditional organization of such a production cell, the robot control acts as a master for the welding control unit. The sequence of the welding operations is coded statically in the robot program. The setup of each welding operation (e.g. welding time, welding current, etc.) is stored on different welding programs in the welding control unit. In a typical sequence, the robot moves to the position of the first welding spot, closes the welding gun and selects the respective welding program of the welding control unit. After that, the robot initiates the welding by triggering a start command of the welding control unit. During welding, the robot is usually not moved. Once the welding process is finalized, the robot moves to the position of the next spot. This traditional way of manufacturing has several disadvantages:

- The sequence of welding operations, the spot locations and the path of the robot arm are implemented in a fixed, static robot program. The easy modification of the program e.g. due to geometrical changes of the assemblies or due to an identified quality problem during operation is not possible online;
- The assignment of the welding parameters on the welding programs is fixed. The robot needs to know which welding program number has to be selected for performing the welding job. This requires high efforts during the ramp-up phase for establishing the collaboration which hampers the exchangeability of the robot and the welding control respectively;
- The welding setup within a program is fixed. If another setup for welding parameters is required e.g. due to changing materials or boundary conditions, the manufacturing process has to be stopped and the new parameters have to be entered manually by an operator;
- The setup of the robot and the welding control usually differs from vendor to vendor and from device to device. Even if the same setup is used, the individual behavior of the devices is different. For that reason, the setup of each device must be (fine) tuned manually according to the application;
- Even if each device is able to provide an assessment of the process quality, the quality check is mostly done separately for each device. No value is available which describes the overall quality result of the respective production step.

In the task-driven environment, the robot and the welding control are replaced by the respective Manufactrons (see Fig. 43). The Handling and the Welding Manufactron are interfacing the robot and the welding control respectively. In order to perform the job, collaboration between the Welding Manufactron and the Handling Manufactron has to be established. The following main steps are required for welding a row of two spots:

1. Start the sequence by moving the tool tips from the parking position to the first spot position (Handling Manufactron);
2. Select proper welding parameters and perform the welding process (Welding Manufactron);
3. Move to the next position (Handling Manufactron);
4. Select proper welding parameters and perform the welding process (Welding Manufactron);
5. Move the gun tips to the parking position (Handling Manufactron).

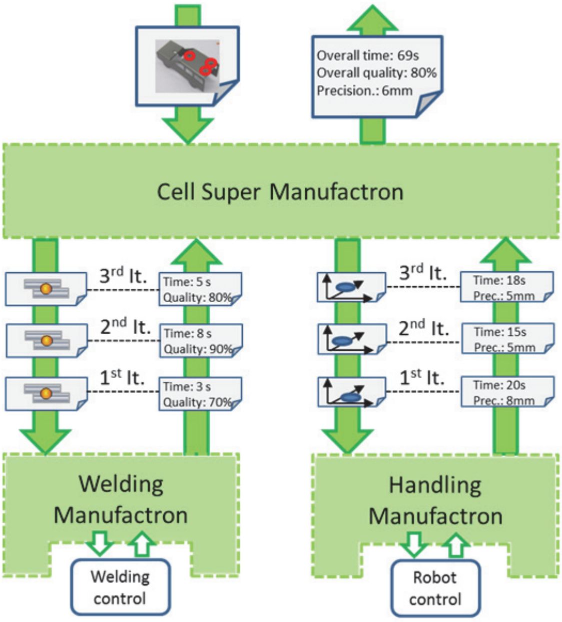


Fig. 43. Schematic design of the manufacturing cell including a Welding and a Handling Manufactron by using the collaboration via task description and Quality Result Documents with three iterations.

Communication is needed when the Handling Manufactron has finalized his job and triggers the Welding Manufactron and vice versa. For this example the communication via a Super Manufactron using the Task Description Documents and Quality Result Documents has been chosen. All the communication goes via a Super Manufactron which acts as a master of communication. It receives a Task

Description Document with the description of the assembly job. After that it queries its knowledge system in order to find a proper method for task execution. In this example, the method contains the specific task descriptions for each subtask for the Welding and Handling Manufactron. Once a proper method is loaded, the Super Manufactron triggers the start of the collaboration by issuing a Task Description Documents to the Handling Manufactron. After performing its job, the Handling Manufactron sends a Quality Result Document back to the Super Manufactron. The Super Manufactron issues the next Task Description Document to the Welding Manufactron and waits for the Quality Result Document and so on. Once the last task description has been send (which is for the Handling Manufactron to move to the stop position) the Super Manufactron assesses all Quality Result Documents received and compiles a document on the overall task performance. In the final step the Super Manufactron issues this document to the manufacturing system.

5.2.8 Discussion

The hierarchical organization and communication model basically relies on the capabilities of the Super Manufactrons, the structure of the Task Description Documents and the Quality Result Documents. The mechanisms for equipment synchronization and the equipment self-description were designed within the framework conditions of the XPRESS project only. Further research is required in order to bring those mechanisms to a broader basis which could then serve as a basis for industrial implementation. For the equipment self-description several approaches exist, e.g. the Device Description Language (DDL) (Chao Chen&Helal 2009) or the Electronic Device Description Language (EDDL) (EDDL 2013). Also IEC 61499 described in Section 2.4 provides related features (IEC 2005).

The Super Manufactrons allow for a hierarchical organization of Manufactrons in the manufacturing system. Similar to the supervisor holons of the ADACOR architecture, optimization and disturbance handling can be improved even further (Leitão 2004). In the context of this thesis, however, the focus is on the hierarchical descriptions of tasks supported by the Super Manufactrons. The hierarchical organization of Super Manufactrons in a manufacturing system allows for the design of ‘abstraction levels’ in the task descriptions (see Section 5.2.1). This means the related Super Manufactron is solely responsible for the management of task descriptions of the sub-

Manufactrons. The Super Manufactron keeps, generates and communicates the task descriptions for the underlying sub-Manufactrons. From ‘outside’ the task descriptions of the sub-Manufactrons are hidden and thus, there is no need to communicate them. Without the Super Manufactrons all tasks required for the production of a product would need to be included in a single Task Description Document. For a large production setup such large documents would hardly be manageable.

There are also some disadvantages when handling the hierarchical organization with Super Manufactrons. Besides general issues described in Section 2.1 some additional challenges with respect to the task-driven approach need to be addressed. Firstly, Super Manufactrons are additional complex entities in a manufacturing system. Trained staff is required in order to implement and maintain them, especially with respect to the features of the knowledge system. As part of the XPRESS project, the system integrators intended to provide Super Manufactrons for dedicated tasks such as panel assembly in aerospace industry, including riveting and positioning tools or car body assembly in automotive industry with a robot and a welding controller. As today’s system integrators are also more familiar with traditional structures and programming, the implementation of Super Manufactrons turned out to be very challenging for them. In addition, passing responsibility from the end users to the systems integrators is in contradiction with the mindset of the end users who want to keep everything “in house”.

The second challenge with respect to the Super Manufactrons was the handling of the various documents. In order to establish a proper collaboration of Sub-Manufactrons coordinated by a Super Manufactron, self-descriptions, data for the synchronization models, task descriptions for the sub-Manufactrons and the respective quality documents need to be consistent. In XPRESS the most sophisticated setup was the collaboration of three sub-Manufactrons (Welding Manufactron, Robot Manufactron and Robot Path Simulation Manufactron). Keeping all data of those Manufactrons consistent by hand was very time-consuming and painful. Consequently, the system developers asked for software tools that assist them in improving the efficiency of the development process.

The main objective of the Task Description Documents and the Quality Result Documents is to provide a general document format for the description of task and quality information. The description concentrates on the general content of these documents. The author of this thesis was responsible for the description as part of the XPRESS project. The details on the structure and technical implementation are described in Hoffmeister (2012) which also provides a detailed data model for equipment synchronization using distributed state machines roughly described in Section 5.2.5.

The Task Description Documents provide a unique task description format for task execution, simulation and emulation. In addition, requests for equipment self-description, state, data and process history can also be formulated. As described above, the task descriptions are not limited to the task descriptions of process tasks on device level, but can contain any task description for the Manufactrons within the hierarchy. This means a broad range of different tasks can be described, e.g. "*Weld two steel sheets of 1.5mm*" for a Welding Manufactron or "*Assemble a car door of car type X*" for a Super Manufactron responsible for a car door assembly cell or "*Simulate a line using equipment X, Y, Z*" for a Simulation Manufactron in combination with a commercial simulation software tool. The Quality Result Documents provide a standard format for gathering task response data from Manufactrons. As a result, the Task Description Documents and Quality Result Documents help to overcome the issue of missing standardization in heterogeneous environments (see Section 2.5).

As described in more detail in Section 6.1.1, Task Description Documents and Quality Result Documents as well as implementations of Process Manufactrons and Super Manufactrons could be integrated in a working production setup in the aerospace industry. This demonstrates the coexistence of the traditional and the task-driven approach. This feature has also been identified as a major requirement for a take-up of advanced technologies in industrial implementation.

As described in the previous section, the success of a wider acceptance of the task-driven approach is related to a certain extend of standardization of Task Description Documents. Different vendors need to come to a conclusion how to describe tasks and how the different requirements of their equipment in terms of the task description can be expressed. In addition, the technical challenges of document handling and managing need to be addressed.

In order to illustrate the potentials of the hierarchical organization and communication model, Table 10 provides a description of their possible contributions to flexible manufacturing systems by a reflection of the requirements identified in Section 3.6.

Table 10. Potentials of the approach of the hierarchical organization and communication model to the requirements of flexible manufacturing systems.

Requirement	Contribution
Flexibility in the introduction of new products	Super Manufactrons enable easy integration of plant sections such as cells or lines. TDDs describe the eBOP and can be duplicated and modified. Documents on self-description of Manufactrons help to select proper equipment for the production of a new product.
Reconfigurability as a reaction to changing demands	Changing demands are expressed only in the TDDs. No re-programming of equipment is required. Super Manufactrons and the synchronization mechanisms allow for logical separation of knowledge domains and avoid master-slave relations. The dependencies between equipment can be limited.
Learning ability of components	No specific contribution
Scalability and extensibility	Super Manufactrons allow for load-balancing of the entire system. TDDs can be easily adapted for system extension.
Re-use of existing systems	The hierarchical structure and the decoupling of knowledge support the integration of legacy equipment.
Deterministic behavior of the system	Super Manufactrons allow for separation of large manufacturing systems into smaller logical entities. To this end, the degrees of freedom for process execution are reduced which finally results in a more deterministic behavior.
Acceptance of existing system boundaries	The hierarchical organization and communication model is implemented according to the traditional factory levels. Synchronization of equipment allows for knowledge domain bridging. No need to merge knowledge domains.
Knowledge-based manufacturing in all levels	Super Manufactrons allow for the encapsulation of knowledge for a certain domain. This allows e.g. system integrators to encapsulate their knowledge on optimal Manufactron orchestration for a certain purpose.
Abstraction of specific knowledge	TDDs provide the access to the knowledge encapsulated in the Manufactrons in a standardized and easy way.
Traceability of product and process quality	QRDs deliver the Manufactron’s response in a standardized way. Super Manufactrons allow for quality data assessment of their domain.

5.3 Quality-oriented workflow

The architecture features for task-driven machines and devices as well as the approach for hierarchical organization and communication described in the previous two sections are related to the shop-floor of a manufacturing system. With the description of workflows in this section we are now approaching the next level of the manufacturing pyramid. Workflows are residing on the MES level and are supposed to orchestrate the Manufactrons in shop-floor level. This is done by bringing the respective Task Description Documents to the Manufactrons in the desired sequence. Furthermore, the workflow mechanism also enables the gathering and persisting of all (quality) data on production issued by the Manufactrons in the Quality Result Documents. At the end of a production cycle a complete documentation of production data is available.

As already illustrated in the discussion of Chapter 4, the mechanism for production orchestration and quality management are tailored to the needs of the industrial partners within the XPRESS project. Due to their philosophy in production operation, restrictions on the dynamic behavior of the entire manufacturing system have been expressed. Those restrictions affect especially the dynamic behavior of the workflow. For that reason, limitations of the structure and the behavior of this part of the architecture are introduced by intention. Nevertheless, the mechanisms are a relevant part of the task-driven architecture and are thus illustrated and discussed within this section.

The main actors in the approach for a quality-oriented workflow are the *Workflow Managers* and *Quality Managers*. Both are software components which build an intrinsic entity of a software agent within the MES.

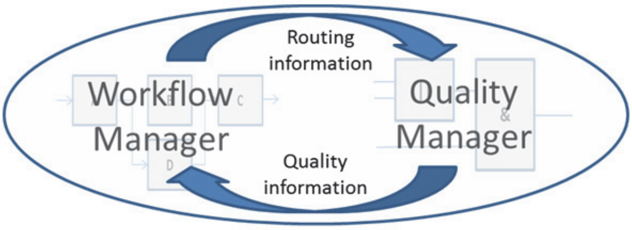


Fig. 44. General structure of Workflow Managers and Quality Managers and information exchange.

During production, Workflow Manager and Quality Manager exchange information in order to enable dynamic routing (see Fig. 44). The task-driven architecture enables to relate process data to product data and vice versa. Furthermore, dynamic routing of the production flow is enabled by assessing the results of a production step and by influencing the subsequent behavior of the production flow in relation to the quality assessment.

In the first section, the production orchestration by Workflow Managers and the quality assessment by Quality Managers are illustrated in detail. Following this, the dynamic routing capabilities provided by the architecture with the help of the cooperation of Workflow Managers and Quality Managers is shown. This section ends with the description of an example and a final discussion.

5.3.1 Production orchestration by Workflow Managers

This section describes the orchestration of the production in a task-driven environment. The Workflow Managers are introduced which are required to deliver the task descriptions to the Manufactrons. After some general aspects, the structure of the Workflow Managers, their generation as well as some aspects of the technical implementation are highlighted.

Workflow Managers

Workflow Managers are agile software agents which reside on the MES level. The purpose of a Workflow Manager is to deliver the task descriptions to the (Super) Manufactrons. The route of the Workflow Managers to the different Manufactrons in the manufacturing system is described by an xml-based description document inside of each Workflow Manager. Besides of simple sequential processing of tasks, Workflow Managers are also capable of processing loops or alternative routes if required. Further description of the purpose of these capabilities is given in Section 5.3.3.

In simple scenarios, Workflow Managers transport a Task Description Document to one Manufactron only. To do so, an instance of a Workflow Manager is generated and equipped with the corresponding Task Description Document. The Workflow Manager instance delivers the Task Description Document to the respective Manufactron. After the quality feedback of the Manufactron, the Workflow Manager instance is not required anymore. Such simple scenarios might be useful for performing simulations jobs for method generation or for the gathering of self-description documents of Manufactrons during the design phase of production.

Workflow Managers are the virtual representation of the *products* within the manufacturing system. In contrast to that, Manufactrons can be seen as proxies of the production equipment on shop-floor and the Super Manufactrons as a product-independent (virtual) entity for the orchestration of underlying Manufactrons Each product type or variant is represented by a corresponding Workflow Manager type. For each product instance, an instance of the respective Workflow Manager type is instantiated at the beginning of the production sequence. The

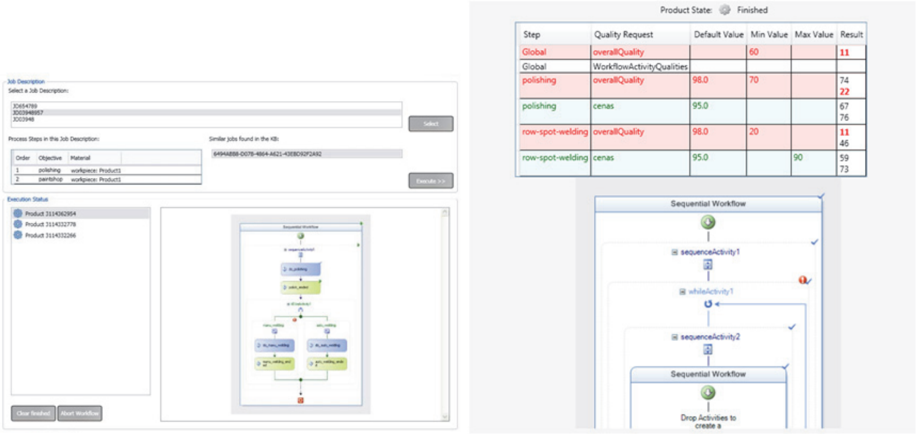
Workflow Manager instance is accompanying the product instance from the beginning of the manufacturing process until the end. In each production step the Workflow Manager releases a Task Description Document to the respective (Super-) Manufactron and waits until the manufacturing tasks have been executed. After that he moves to the next Manufactron in the production sequence.

In summary, the Workflow Managers can be seen as the generic communication mechanism in the task-driven manufacturing system for the delivery of Task Description Documents. The workflow mechanism is used independent from the task type and the purpose of the system setup. During the design phase, simulation tasks for method generation or self-description requests for Manufactron capability query can be communicated. Furthermore, the communication flow within virtual setup of entire manufacturing environments can be emulated either in advance or in parallel to the real production. In real manufacturing environments in the production phase, Workflow Managers are used for the production orchestration of the entire production system.

Workflow Manager generation and workflow management

Workflow Managers are created on ERP level by the so-called *Production Execution System* (PES). This software tool is coupled to the factories scheduler which receives the order data for the products to be produced. For the instantiation of a Workflow Manager, the PES queries its database for a proper Workflow Manager template. The template is a generic description for the production of a *product type* and basically consists of the eBOP for the product manufacturing (see Section 4.4). Workflow Manager templates can consist of variables which are filled once the concrete *product instance* is generated. If for example a car door shall be produced, the Workflow Manager template consists of the description of the materials, quality requirements, required equipment and operation sequence of the manufacturing process. The variables might be the color of the car door frame or specific interior components which differ from product instance to product instance, but do not influence the manufacturing process as such. After that the template is instantiated and filled, the Workflow Manager agent starts the execution of the manufacturing job.

All instantiated Workflow Managers are residing in an “environment” called *Workflow Execution System* (WES). The WES is implemented as a software component which is capable for Workflow Manager movement, tracing, tracking and destroying.



Select a Job Description:
JD654789
JD03948957
JD03948

Process Steps in this Job Description:

Order	Objective	Material
1	polishing	workpiece: Product1
2	paintshop	workpiece: Product1

Similar Jobs found in the KB: 6494ABB8-D07B-4864-A621-43EBD92F2A92

Product 3114362954
Product 3114332778
Product 3114332266

Product State: Finished

Step	Quality Request	Default Value	Min Value	Max Value	Result
Global	overallQuality		60		11
Global	WorkflowActivityQualities				
polishing	overallQuality	98.0	70		74 22
polishing	cenas	95.0			67 76
row-spot-welding	overallQuality	98.0	20		11 46
row-spot-welding	cenas	95.0		90	59 73

(a) (b)

Fig. 45. Graphical user interface for workflow management: (a) Workflow Manager generation on the basis of templates stored in a database; (b) Workflow Manager tracking and process step assessment (Gigler *et al.* 2012, published by permission of Fraunhofer IPA).

A *Directory Service* is available which is used for registering and resolving network addresses and Manufactron names. Fig. 45 shows the implementation of graphical user interfaces for Workflow Manager and Manufactron monitoring. (Gigler *et al.* 2012).

According to the specification of the Workflow Execution System described above, only one (global) WES is existing in a factory which is solely responsible for controlling all workflow processes. However, for dedicated applications, two variants for workflow managing do exist. In order to enable load balancing and (restricted) asynchronous workflow behavior, the architecture allows the implementation of distributed Workflow Execution Systems. In this concept, a global WES on factory level is supported by several Sub-WES which reside on machine level. The global WES delegates sub-tasks to the Sub-WES which then orchestrates the related Manufactrons independently from the global WES. By this, pipelined machines can be supported and the robustness of the whole system is improved. (Almeida *et al.* 2011). The second variant in workflow management is relevant in industrial setups using Automated Guided vehicles (AGV’s). In contrast to other production resources which are located at fixed positions within the production environment, the AGV’s are agile resources. As all other resources, AGV's are also implemented as Manufactrons which receive task descriptions for fulfilling a dedicated job. In this case, the task is simply transporting components from position *A* to position *B* why AGV-based Manufactrons are called *Transport Manufactrons*. However, due to their agility and specific features, the potential occupation of AGV's is not only related to their intrinsic capabilities but also to other boundary conditions such as their actual position, battery load or remaining capacity, battery charging time, etc. As a matter of fact, AGV-based transport systems usually consist of a fleet of AGV's. This allows for fulfilling parallel transporting tasks and to provide the opportunity that one AGV is taking over the job of another AGV in case of AGV malfunctions or maintenance. For such complex scenarios the Workflow Manager concept is extended by the *Transport Managers*. The Transport Managers are supposed to control and to supervise a complete fleet of AGV-based Transport Manufactrons. To do so, the Transport Manager consists of the factory layout as well as the actual position and state of each Transport Manufactron within the factory. The Transport Manager is coupled to the Workflow Execution System which forwards a transporting task to the Transport Manufactron. The Transport Manager identifies the most capable Transport Manufactron and dispatches the transporting task. Thus, the Transport Manager is responsible to track the whereabouts of the Transport Manufactrons, for the calculation of potential transporting scenarios and for the orchestration of the AGV's for task execution. (Almeida *et al.* 2010).

5.3.2 Managing quality by Quality Managers

The Quality Managers are supposed to correlate process-specific (quality) data with the related product entity. This aims for a 100% quality control and documentation of all production processes and steps which are involved in the manufacturing process of a product. While nowadays process quality is often not correlated to the quality of the final product and the documentation of quality during production is often incomplete, the Quality Managers are storing and assessing the feedback on quality-related data from each production process (see also REQ 10 "*Traceability*" on page 62). Quality Managers are created in parallel to the Workflow Managers for each product entity to be produced. Quality Managers are only logically separated from the Workflow Managers, but Workflow Manager and Quality Managers build one software agent. As the Workflow Managers are more the active part of a software agent (having routing capabilities, actively bringing Task Description Documents to the Manufactrons, etc.), the Quality Managers can be seen as the passive part which comes into play when a production step has been finalized.

Quality data gathering

Quality Managers gather the Quality Result Documents issued after each task execution by the Manufactrons. At the beginning of a production job, the Workflow Manager issues a Task Description Document to the respective Manufactron including the requests on quality data (Fig. 46a).

```
<qualityRequest>
   <!-- Overall quality value -->
   <qualityRequestItem>
     <name>overallQuality</name>
     <type>double</type>
     <defaultValue>0</defaultValue>
   </qualityRequestItem>
   <!-- Deviation in X position -->
   <qualityRequestItem>
     <name>deltaPosX</name>
     <type>double</type>
     <defaultValue/>
   </qualityRequestItem>
   <!-- Deviation in Y position -->
   <qualityRequestItem>
     <name>deltaPosY</name>
     <type>double</type>
     <defaultValue/>
   </qualityRequestItem>
   <!-- List of WF qualities -->
   <qualityRequestItem>
     <name>WFActivityQualities</name>
     <type>list</type>
     <defaultValue/>
   </qualityRequestItem>
   </qualityRequest>
```

(a)

```
<QualityResultDocument/>
  <!-- Calculated overall quality -->
  <qualityResultItem>
    <name>OverallQuality</name>
    <type>double</type>
    <value>95.5</value>
  </qualityResultItem>
  <!-- Actual deviation in X position -->
  <qualityResultItem>
    <name>deltaPosX</name>
    <type>double</type>
    <value>3.0</value>
  </qualityResultItem>
  <!-- Actual deviation in Y position -->
  <qualityResultItem>
    <name>deltaPosY</name>
    <type>double</type>
    <value>2.3</value>
  </qualityResultItem>
  <!—Act. WF qualities -->
   <qualityResultItem>
     <name>WFActivityQualities</name>
     <type>list</type>
     <value>5.6;9.3;1;4.2</value>
   </qualityResultItem>
</QualityResultDocument>
```

(b)

Fig. 46. (a) Quality request within a Task Description Document and (b) Quality Result Document issues after the task execution.

After the execution, the Manufactron issues a Quality Result Document to the Quality Manager which consists of the requested quality items (Fig. 46b). As the Quality Managers accompany the Workflow Managers from production step to production step, at the end of the production cycle the Quality Managers contain the entire quality data of each production process related to the produced product/variant. This data can then be stored for documentation and further analysis purposes.

Quality data assessment

Gathering and storing the quality data is only one of two major features of the Quality Managers. The second feature is the capability for assessing the quality data inline with the production process. Quality Managers can be equipped with embedded rules and algorithms for assessing the result of a production process based on the feedback of the respective Manufactron and for drawing conclusions. Quality Managers do usually not care about the quality assessment methods for specific production processes, e.g. calculate the quality of a welding spot based on a process signal analysis. Such dedicated calculations are the responsibility of the respective Manufactrons. However, the Quality Managers are using such data in order to estimate the influence of each single process result on the overall quality of the product. This can be achieved by coupling the Quality Managers with the Workflow Managers for which the proposed architecture provides a basis. By coupling Quality Managers and Workflow Managers, both have access to the complete product description and the quality requirements. In addition to that, by accompanying the product during its evolution from process step to process step in the production chain, Quality Managers do have access to the quality data history. By this, Quality Managers are able to assess quality not only locally at the actual process step, but globally over the entire production chain. This feature is important due to the error propagations in production. If for example, minor quality faults occur at the process steps n-3, n-2 and n-1, a local assessment of the quality would detect a significant lack of quality as they are individually not an issue. However, the sum of faults could lead to a drastic drop of quality of the entire product. Furthermore, only if the history of quality of the previous production steps is identified and the requirements on incoming quality of the subsequent processes are known, proper measures for process and quality regulation can be derived.

5.3.3 Dynamic routing of production flow

Workflow Managers and Quality Managers always build a pair and are coupled internally for data exchange. By this, Workflow Managers are able to react dynamically on the assessment result of the Quality Managers. If for example, insufficient quality has been recognized by the Quality Manager after a certain production step, the Workflow Manager is able to change the route of the production steps (which is implicitly contained in the task descriptions) in order to guide the product e.g. to a rework station. To do so, the following data is exchanged between Workflow Managers and Quality Managers:

- Data from Workflow Managers to Quality Managers:
  - Routing information;
  - Information on equipment;
  - Quality requirements.
- Data from Quality Managers to Workflow Managers:
  - Quality data of the last production step;
  - Quality data with respect to the entire product;
  - Alarms, error messages and other events.

```
<cf:WhileActivity>
  <cf:WhileActivity.Condition>
    <cf:RuleCondition Condition="(!QrdResults.ContainsKey("a5") ||
                                   QrdResults["a5"] < 50) &&
                                   QrdResults["a3"] < 50"/>
  </cf:WhileActivity.Condition>
  <cf:SequenceActivity Name="seq1">…</cf:SequenceActivity>
</cf:WhileActivity>
```

(a)

```
<cf:IfElseActivity>
  <cf:IfElseBranchActivity Name="manu_welding">
  <cf:IfElseBranchActivity.Condition>
    <cf:RuleCondition Condition="!IsAvailable("rsw2") ||
                                  IsAvailable("rsw1")"/>
   </cf:IfElseBranchActivity.Condition>
   </cf:IfElseBranchActivity>
</cf:IfElseActivity>
```

(b)

Fig. 47. Examples of control flow conditions: (a) While activity and (b) if-else activity (see also Gigler *et al.* 2012, published by permission of Fraunhofer IPA).

The Workflow Manager is now able to react on the quality feedback of the Quality Manager. To do so, the Workflow Manager's control flow contains conditions which influence the work flow with respect to the quality result (see Fig. 47). Changes in the control flow can be done for different purposes. In case of quality issues identified by the Quality Managers, one or more production steps might be repeated, the product can be re-routed to a dedicated rework station or in case of total quality loss, the product can be sorted out of the manufacturing process. The capability for re-routing can also be used for load balancing or in case of the unavailability of production equipment.

The physical material and subassembly transport from unit to unit can either be done by conventional transport or feeder systems or by dedicated transport Manufactrons which are interfacing AGV's. The latter provides the possibility of more flexibility in task execution by e.g. finding alternative transportation routes, identification of bottlenecks or by better load balancing of production equipment.

5.3.4 Quality-oriented workflow in practice

This section provides some information on the implementation of the quality-oriented workflow in industrial practice. It concentrates on two cases which are considered the most relevant ones: The first is the setup of the environment in which the various entities for workflow orchestration are instantiated. The second case illustrates some details on the instantiation and programming of the Workflow and Quality Manager and their templates. The efforts for industrial implementation of the workflow approach are based on the personal experience of the author of this thesis and on the information provided in Gigler *et al.* (2012). Due to confidentiality reasons, more details on the technical implementation of the various system entities cannot be provided.

Setup of the workflow environment

As described above, the environment in which the workflows are residing is called Workflow Execution System (WES). The WES was implemented using the Windows Workflow Foundation (WF) (see also Gigler *et al.* 2012). The WF is a Microsoft tool designed to assist companies in coordinating their processes (Scribner 2007). The transferability of WF to an industrial environment was investigated as part of XPRESS.

The WES was implemented once and was used for several production setups within the XPRESS project. Only minor adaptations or extensions were required in order to tailor the WES to the different needs. In the test cases the WF scaled well and its availability was high. Furthermore, the workflow concept is seen to be very extensible and well suitable for customization (Gigler *et al.* 2012). However, as the principle production philosophy of the companies involved was quite similar, a general statement on the generic use of the WES in other production environments cannot be given.

In order to make the WES working, a number of services need to be provided. Besides the Directory Service mentioned above, tools for workflow monitoring and debugging need to be implemented. Especially the debugging of the (distributed) behavior of the workflow turned out to be a critical issue. Different time settings of the involved servers, lead times of workflow operation as well as different response times due to varying network traffic are associated with great effort regarding bug finding and fixing, even in relatively small system setups. Therefore, it is assumed that extensive efforts are needed in order to make the system work in larger setups. The disadvantage of distributed systems with respect to greater efforts with monitoring and debugging was also mentioned as one blocking point for the take-up of advanced technologies in industrial practice.

Instantiation and programming of the Workflow and Quality Manager

Each product type or variant within a task-driven manufacturing system is represented by a Workflow Manager and Quality Manager template. As soon as a certain product shall be produced, the respective template is identified and a workflow entity is instantiated. During the instantiation, the variable data for the production of the product is set.

Template identification, workflow instantiation and data insertion is performed automatically by the PES provided that a proper connection to an ERP system is available. A test case in a paint shop implemented within the XPRESS project is described in Section 4.6. The efforts for industrial implementation depend on data availability and on ERP system interfacing. As most of these systems provide proper interfaces and as this work only needs to be done once in a manufacturing system, the overall effort is assumed to be relatively low.

In contrast to the workflow instantiation, the generation and maintenance of the Workflow Manger and Quality Manager templates are more sophisticated. The templates describe the route which a certain product is supposed to take within the factory. The routes can dynamically be influenced by the Quality Managers which assess the result of a task-driven production step. At first glance this concept seems to be relatively simple. Using WF, workflows can be programmed quite easily. Additionally, it enables dynamic behavior of workflows as WF also allows for programming of simple conditions in the program flow. For more sophisticated rules, external algorithms, e.g. programmed in C#, can be embedded in WF code. In addition, WF code is XML-based and can thus easily be edited and duplicated in order to “reuse” proved workflows for similar products and production setups (see also Gigler *et al.* 2012). A more detailed analysis exposes several weak points of the approach. Similar to the issues related to the management of Task Description Documents identified in Section 5.2.6, the manual handling of the workflow code is time consuming and error-prone. However, the major issue is to implement dynamic workflow behavior. The design of workflows for dynamic resource allocation (e.g. bottleneck management using redundant equipment) and disturbance handling (e.g. malfunction within the Manufactrons or the repetition of a production step after an identified quality problem) is generally possible in WF. However, the respective WF code gets very complex. All eventualities of potential dynamics within the entire production flow need to be known in advance and need to be coded into a single workflow document. This hampers the ad hoc behavior of the system and also hinders the adaptation of workflows to different production setups.

5.3.5 Example: Two-cell production line

The cooperation of the Workflow and Quality Managers is illustrated with the simple case of two production cells and three products within the production line. Fig. 48 on the next page shows the setup of the scenario. Two production cells are arranged in a sequential order. In Fig. 48 (a) two subassemblies are arriving at production Unit 1. They are accompanied by a pair of Workflow Manager and Quality Manager (WfM #1). The Workflow Manager is issuing the description of the task for the production step to Unit 1 which is performing the task according to the instructions of the task description (#1). Unit 2 is not involved in this step. Fig. 48 (b) shows the next step. Unit 1 finalized its work on Product No. 1 (WfM #1) and respectively, the Workflow Manager and the Quality Manager proceed one step ahead. The Quality Manager gathers all relevant data on the finalized step from Unit 1 (#2). Next, the Workflow Manager issues the next task description to the next unit (Unit 2) (#3a). At the same time, product No. 2 arrived at Unit 1 and the responsible Workflow Manager (WfM #2) issues the task description for product No.2 to Unit 1 (#3b).

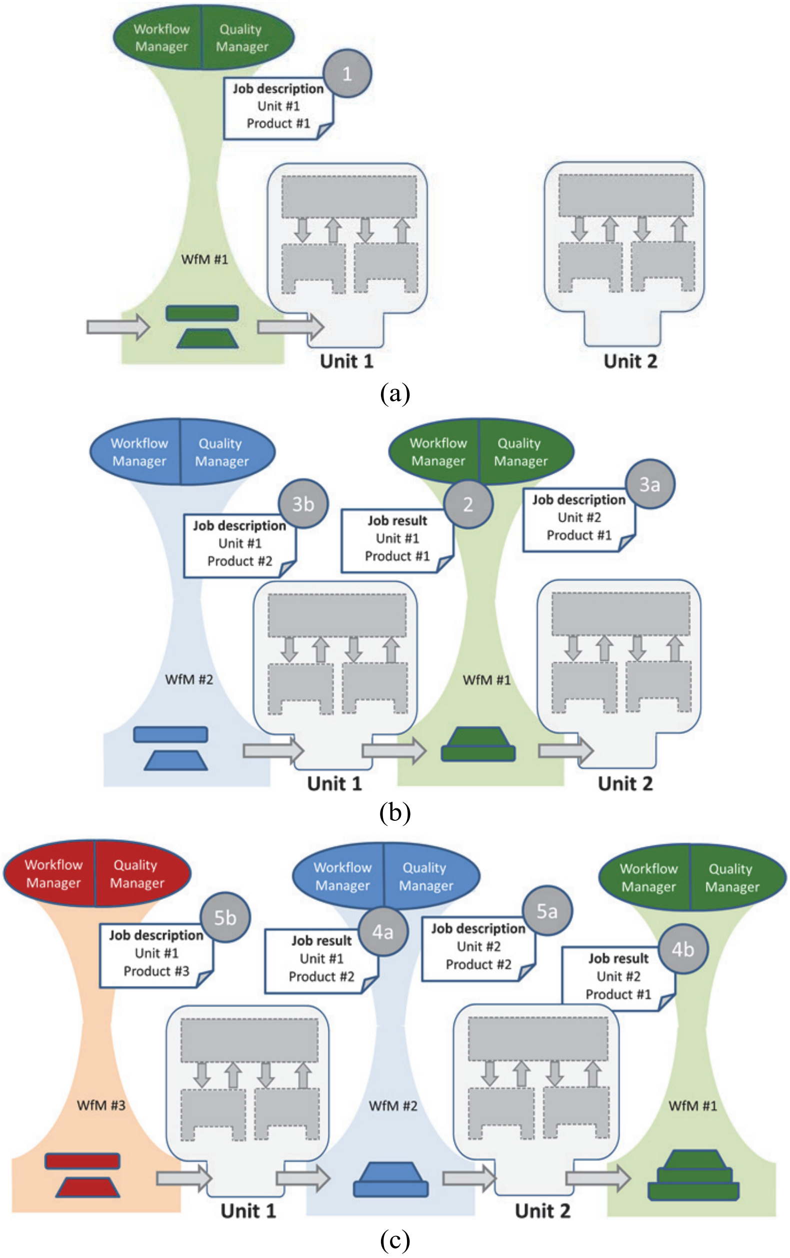


Fig. 48. Scenario with two production units and the production of three products. (a) Product 1 arrives at Unit 1; (b) Product 1 moves to Unit 2 and Product 2 arrives at Unit 1; (c) Product 1 leaves Unit 2, Product 2 passes Unit 1 and arrives Unit 2, Product 3 arrives at Unit 1.

In Fig. 48 (c) Product No. 1 (WfM #1) is finalized. All quality data from Unit 1 and Unit 2 are aggregated by the respective Quality Manager (WfM #1) (#4b). Product No. 2 is arriving at Unit 2 and Product No. 3 is arriving at Unit 1. By this mechanism, all relevant data on the production process of each product is gathered. At the end of all production steps, a comprehensive collection of process data in relation to each product produced is available.

5.3.6 Discussion

With the introduction of Workflow Managers and Quality Managers a standard mechanism for the communication of Task Description Documents to the Manufactrons and for the gathering of quality information is proposed. The approach also allows for a dynamic production flow in case of disturbances by coupling Workflow and Quality Managers. The approach is tailored to the needs of the industries involved in the XPRESS project. Restrictions on the dynamic behavior of the related manufacturing systems limit the degree of freedom of decision taking and product routing. As part of XPRESS, the test cases the approach worked well. After the implementation of the Workflow Execution System only minor adaptations to the various environments of production were required. Moreover, system scalability for the workflow instantiation was good and debugging tools assist programmers during the development process (Gigler *et al.* 2012). Limits to agent instantiation and system development were mentioned as blocking points for the industrial implementation of agent-based manufacturing systems (see Section 2.5). The deterministic behavior of production execution reached by a relatively fixed workflow sequence and prior production simulation and emulation helped to reduce the averseness of the involved industries to agent-based manufacturing systems. The dynamic product routing capabilities and disturbance handling met the partners' expectations. The Quality Managers allow data gathering and assessment along the entire production chain.

Taking a closer look at this, there are also some limitations of the approach. While sequential production setups are relatively simple to address, more complex setups including parallel tasks and sophisticated disturbance handling lead to complex Workflow Managers. As a consequence of this, proper software tools and skilled personnel are required in order to enable an effective generation of Workflow Manager templates. Further research is required in order to estimate the applicability of the proposed approach for production setups which go beyond the scope of the requirements for dynamic routing expressed in Chapter 3. Tentatively, a limitation of the approach is seen in case of large production setups and highly dynamic manufacturing environments. Table 11 shows the potentials of the quality-oriented workflow with respect to the requirements.

Table 11. Potentials of the quality-oriented workflow with respect to the requirements of flexible manufacturing systems.

Requirement	Contribution
Flexibility in the introduction of new products	Workflow Managers allow for a standardized communication of task descriptions to the Manufactrons. This enables distributed simulation which can help to reduce ramp-up time.
Reconfigurability as a reaction to changing demands	The Workflow Manager's routes are expressed in text files which can be adapted to the new demands. Dynamic routing extends the system's flexibility.
Learning ability of components	Workflow Managers receive quality information from the Quality Managers. Algorithms can be integrated which allow Workflow Managers to draw conclusions on the basis of the quality feedback.
Scalability and extensibility of the system	As the routes are expressed in text files, Workflow Managers can be extended if an extension of the manufacturing system is required. Distributed Workflow Execution Systems allow for system scalability.
Re-use of existing systems	No specific contribution
Deterministic behavior of the system	This risk of non-deterministic behavior can be reduced by implementing restrictions on the dynamics or by introducing sub Workflow Execution Systems.
Acceptance of existing system boundaries	The quality-oriented workflow approach takes the classical hierarchical layer structure into account. The Quality Managers can support reasoning in case of errors, malfunctions, etc.
Knowledge-based manufacturing in all levels	Workflow Managers and Quality Managers can be equipped with explicit knowledge on routing and quality assessment. This enables integration of respective know-how into MES systems.
Abstraction of specific knowledge	No specific contribution
Traceability of product and process quality	Quality Managers provide a standardized mechanism for gathering and assessing the process quality and assign it to the respective product.

6 Validation and industrial relevance

Within this chapter, the usability of the proposed architecture for a task-driven manufacturing system is discussed. The basis for the validation are three industrial demonstrators in aerospace, automotive and electrical industry as well as one software demonstrator which implements a virtual manufacturing environment for photovoltaic wafer production. For each demonstrator the setup is described and the various test scenarios are discussed in more detail. After that, the validation is done by reflecting the demonstrator's usability with respect to the goals of this thesis described in Section 1.3[8]. The tests have been carried out within the final demonstration scenario validation of the XPRESS project. All demonstrators have been realized with the contributions of most of the XPRESS partners. The author of this thesis was responsible for the overall planning of the Demonstrators within a "Demonstrator Roadmap" (Peschl 2010). Following this, the demonstrators have been developed according to the architecture described within this thesis. The demonstrator setup and the respective results have also partially been presented at the final XPRESS Status Colloquium (Hoffmeister&Peschl 2012). In addition, the usability of the architecture is also reflected against the processes and process flows of the three related industries: Additive Manufacturing, heat treatment and sheet metal industries. This chapter ends with a discussion of the technical and scientific achievements of the architecture.

6.1 Validation by the XPRESS Demonstrators

The validation of the task-driven manufacturing system is done in the context of four demonstrators in different industrial areas. Demonstrator #1 focuses on the aspects of quality data generation, gathering and assessment in the aerospace industry. It also shows how task-driven technology can be introduced in existing real production environments. In Demonstrator #2, the capabilities for distributed simulation are shown by the example of robot path planning and welding parameter simulation in automotive industry. Demonstrator #3 is the one in which the most of the proposed technology developed within this thesis has been integrated. It focuses on multi-variant production in the electrical industry.

---

[8] The validation results, in particular the degree of fulfillment and the quantifications, are based on the evaluations reported in Peschl (2011), in Goncalves (2011a) and in Preusse (2011). Due to confidentially reasons, not all data can be disclosed within this thesis.

Finally, Demonstrator #4 is a pure software demonstrator. Its purpose is the demonstration of scenarios for distributed simulation and emulation, on factory planning processes and on factory layout optimization and bottleneck identification.

6.1.1 Demonstrator #1: Aerospace industry

The first application for the validation of the task-driven approach is the fuselage manufacturing in the aerospace industry. Such applications are characterized by a huge amount of variety of products. Each airline has individual requirements on the construction and furnishing of their airplanes. For example, the number and the position of seats, cabins or the lavatories are individually chosen which also influences the construction of the fuselage. The position of vertical elements ("Ribs") and horizontal elements ("Stringers") as well as the number and position of the rivets have to be designed according to the needs of the airlines. For that reason, almost all airframes are unique.

Fig. 49. Working environment in aerospace industry: Operator panel with control systems (top left), riveting head (top right) and factory shop-floor with riveting machines and fuselages (bottom).

Fuselage shells are huge components with dimensions up to 7 x 10 m (see Fig. 49). The fuselage shells are produced using aluminum, titanium, carbon-fiber-reinforced plastic (CFRP) and glass-fibre reinforced aluminum (GLARE) with different thickness and various material combinations. For joining the skin plate on the ribs and stringers, usually rivets are used. One fuselage shell can consist of several thousands of rivets. For manufacturing, the shells are fixed on frames and a riveting head is moving to the desired joining positions. For setting a rivet, the following process steps have to be performed: 1.) clamping, 2.) drilling, 3.) sealing, 4.) riveting. The most important quality criteria of the riveting process are a.) precision of the position of the rivet, b.) strength of the joint, c.) height of the rivet head and d.) diameter of the countersink.

The scope of this demonstrator is quite wide as it includes most of the objectives of this thesis (see Section 1.3). By the logical separation of the riveting machine in the riveting head and the positioning unit and by covering both by a “*Riveting Manufactron*” and a “*Panel Assembly Manufactron*” respectively, the basis of the task-driven approach is built (see Fig. 50).

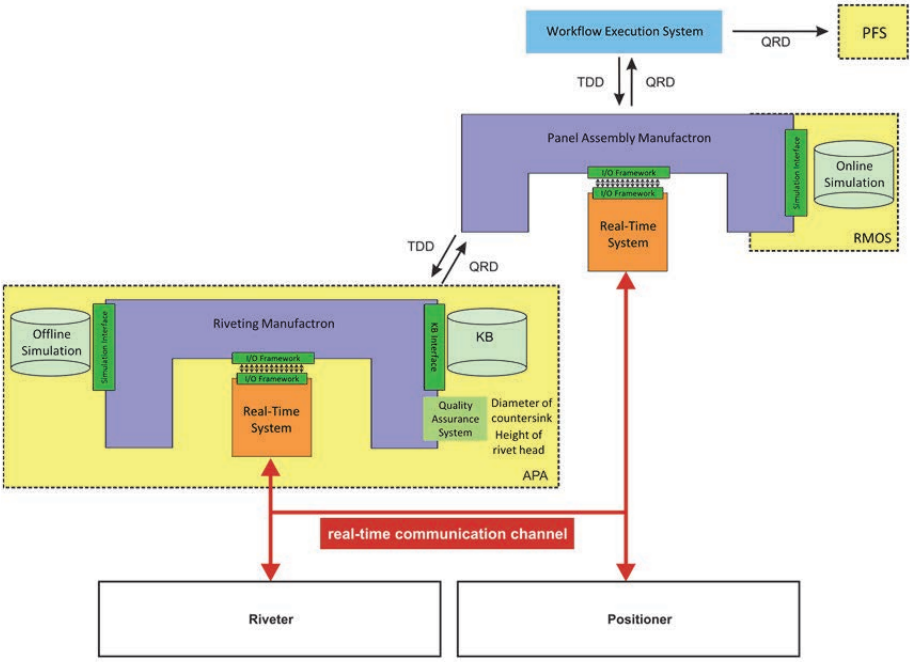


Fig. 50. Setup of the task-driven environment for Demonstrator #1 in aerospace industry with Panel Assembly Manufactron and Riveting Manufactron, each of them interfaced to the respective real-time system (Eickhorst&Trostmann 2012, published by permission of Fraunhofer IPA).

The Riveting Manufactron consists of a knowledge base including parameter data for the riveting process and an offline simulation component for fast parameter finding. Furthermore, a quality assurance system is included which allows for the automatic detection of quality faults by an optical measurement of the rivet's position. The Panel Assembly Manufactron acts as a Super Manufactron for the Riveting Manufactron. It consists of an online simulation unit which assists the operator for easy, safe and collision-free moving of the riveting head in very complex geometries. It is also responsible for the control of the positioner during operation. Thus, the Panel Assembly Manufactron has two roles: a.) The Super Manufactron role which is responsible for the coordination of the Riveting Manufactron and b.) The Process Manufactron role which has the knowledge for the simulation and the execution of the positioning process. Both Manufactrons are implemented according to the specification described in Section 5.1.3 and are communicating with the standardized Task Description Documents and Quality Result Document described in Section 5.2.2 and Section 5.2.3. The interfacing with the real-time systems (PLC S7-400 and SINUMERIK 840D) is done with the approaches described in Section 5.2.5. The quality data is gathered by Quality Managers according to the description in Section 5.3.2 and displayed in a virtual environment ("*Production Feedback System (PFS)*") using CAD data from the CATIA system.

The major novel aspect within this scenario by the proposed architecture is the separation of process knowledge and process execution. Traditionally, the rivet type and the related riveting parameters (e.g. rate of feed, upset and clamping force) to be executed are directly connected to the position of the riveting head. When the riveting head reached the desired position on the panel (which is equivalent to a certain position within the NC program sequence), the riveting process has been executed by loading a riveting program identified by a unique identifier. This means that the riveting program to be executed was directly linked with the position of the riveting head. As a consequence, the program code needed to be adapted whenever the riveting task changed (e.g. due to different material combinations in geometrically identical fuselages) or whenever a different position needed to be approached (e.g. caused by modifications in the geometry). In contrast to that, the task-driven approach separates the riveting program and the position of the riveting head. To do so, the Riveting Manufactron's knowledge base consists of all riveting task descriptions and the respective riveting parameters required for the process execution. The knowledge base can be extended by new tasks without the modification of a NC program code. When the riveting head has reached the desired position on the fuselage, the Riveting Manufactron receives a respective task description within a Task Description Document and executes the riveting process. The task description is automatically derived from the CAD data of the fuselage available in the CATIA system. If a task needs to be changed due to a relocation of a rivet position or due to a modification of the material composition no modification of the position unit's NC program is required[9]. This adds flexibility both to the managing of the methods for task execution and to the execution to the riveting process itself.

Another novel feature by the implementation of the proposed architecture in this application area is the unique method for quality data gathering for both, Riveting and Panel Assembly Manufactron. The Riveting Manufactron assesses the process quality after execution with the help of its internal quality assurance system and delivers the quality values in form of Quality Result Documents to the Panel Assembly Manufactron. The Panel Assembly Manufactron then synchronizes the quality data with the task data (riveting head position) and issues the consolidated information to the Production Feedback System via the Quality Managers.

Table 12 on the following page summarizes the benefits for the implementation of the task-driven approach in the demonstrator in the aerospace industry and provides an estimation on the degree of fulfillment of each objective. Most of the objectives aimed during the validation test of the demonstrator have been fulfilled. It could be demonstrated, that the task-driven approach already allows for benefits even if not the entire architecture has been implemented (due to the fact that the demonstrator is located in a real production environment, only one cell has been equipped with the novel approach. Furthermore, the workflow component is not completely integrated). The main issues figured out during the implementation of the architecture and the tests of the implementation are related to the increased complexity of the system. With the separation of the riveting machine into two logical units, additional entities have been introduced in the manufacturing system for which a reliable collaboration needed to be implemented.

---

[9] A relocation of the rivet position also requiress a dynamic calculation of the (collission-free) path the riveting head needs to follow to the desired position. This is done be a simulation in advance of the execution. This feature would be assigned to the Panel Assembly Manufactron, but has not implemented within this Demonstrator.

Table 12. Validation of the task-driven approach in aerospace industry.

Goal	Fulfillment	Description	Quantification
Decrease production ramp-up time	●	The embedded simulation in the Riveting Manufactron in combination with its knowledge system reduces the efforts for setting up the riveting process.	–* (Up to 75% time reduction using virtual commissioning (Reinhart & Wünsch 2007)
Methodology for multi-variant production	◕	Knowledge on riveting process execution and positioning is separated which leads to a more general use of the respective knowledge.	–*
Easy implementation in the industry	●	The task-driven approach has been integrated into an existing production environment at the premises of Premium Aerotec in Nordenham, Germany for supporting the production of the A380 super jumbo.	Integration effort for this Demonstrator was approximately 12 man months.
Generate and exploit vendor's knowhow	◕	Process knowhow on optimal riveting process parameters is embedded in the Riveting Manufactron. This improves the competitiveness of the Riveting Manufacturer's vendor as they are able to provide advanced services.	Not applicable
Documentation of production data	●	The Riveting Manufactron consists of a quality assurance system for riveting process inspection. The quality data is gathered for the entire riveting process.	100% quality data tracking and assessment

* Quantification available within the XPRESS Consortium, but not publishable.

Fulfillment: ● = fully achieved; ◕ = mainly achieved; ◑ = partially achieved; ○ = not achieved

The maintenance of them and especially the distributed intelligence and responsibility led to increased efforts. Furthermore, due to the incomplete integration of the workflow component, a lot of efforts in manual data handling (e.g. Task Description Documents, NC programs, Quality Result Documents) were required. Besides the technical challenges, also other issues needed to be taken into account: Even if the end user was very interested in the results of the implementation, the implementation itself has not been done by the end user but by a system supplier and by an industry-related scientific partner. Daily work load and different requirements on technical backgrounds hampered the full integration of the end user in the technical implementation. As a consequence, additional efforts in the training of the end user for further developments and maintenance of the system are required. Furthermore, it turned out that the system supplier has a key role in such a constellation: Either the supplier is willing to open all required interfaces to the systems in order to allow the required data access or the entire integration needs to be done by the supplier itself. The latter is, however, in contradiction to the end users intention for having the possibility of full data access and adaptability of the system.

The implementation of the both Manufactrons has been done on the basis of the Manufactronic Framework (see Section 5.1.4). The main efforts for the adaptation to the Demonstrator were required in a) interfacing the real-time systems (8 man months), b) building of the knowledge system including the task to method transformation algorithms (12 man months), c) describing the tasks and managing the respective Task Description Documents (8 man months) and d) the implementation of the quality assurance system for riveting processes (10 man months)[10].

6.1.2 Demonstrator #2: Automotive industry

The second case study focuses on the reduction of the ramp-up time for manufacturing systems in automotive industry. In car body shell work, high efforts have to be spent in order to obtain optimal parameters for the various processes. Handling processes carried out by robots and welding processes, in particular resistance spot welding, are two processes which are of great importance.

Nowadays, robot path generation is done either manually using teach-in technology or by using robot path simulation tools such as Process Simulate (Siemens 2013) (Fig. 51 lower picture on the left.). The simulation result is a (semi-)optimized robot program, directly implemented e.g. in KUKA Robot Language (KRL) (see also KUKA 2013 and Mühe *et al.* 2010). Parameter finding for the welding process is even more complex and time consuming. A car bodyshell is composed of a huge amount of different material types and thicknesses. In average, about 5.000 welding spots are used to guarantee the required stiffness of the frame.

---

[10] The numbers reflect the estimated efforts required for the respective work within the XPRESS project. This included research, implementation and integration activities. The related partners estimate a significant reduction of the efforts for subsequent installations due to available experiences and technologies.

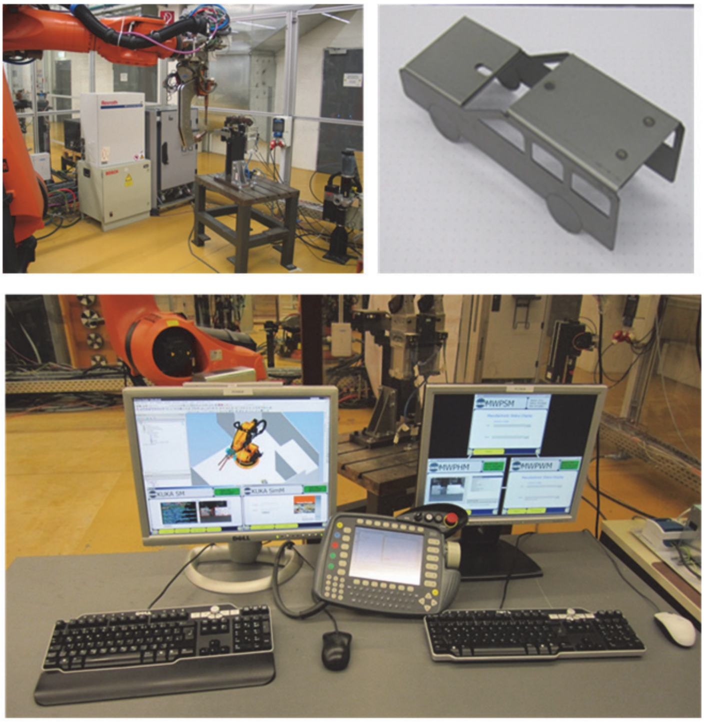


Fig. 51. Setup of the Demonstrator environment in the automotive industry: Typical robot cell (top left), test product (top right) and simulation environment (bottom).

For each welding spot, proper parameters such as welding current level, welding time and electrode force are required. Nowadays, such parameters are mostly generated by performing manual time consuming welding tests in advance of the production or by try-and-error methods. To do so, the influences of the various welding parameters on the welding quality for all material combinations are tried out in an approximation procedure using test metal sheets. To overcome such time and cost intensive methods, the manual parameter finding has to be reduced.

In this context, the demonstrator aims for a (semi-)automated generation of robot paths, for an automated generation of welding parameters by the integration of welding simulation software based on the Finite Element Method (FEM), on reusing simulation data and on an online interfacing of a robot and a welding control unit for welding job execution. To do so, the demonstrator implements the design illustrated in Fig. 52. The general design is similar to the design of the example illustrating the cooperation between Manufactrons in execution mode in Section 5.1.5. The collaboration between a Welding Manufactron and a Handling

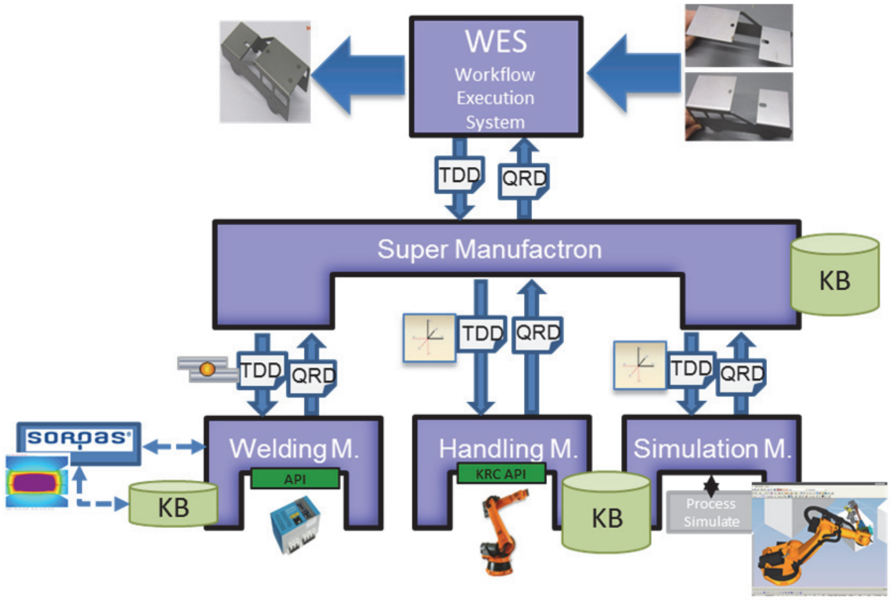


Fig. 52. Manufactronic design of Demonstrator #2 for distributed simulation in automotive industry (Fedrowitz 2012, published by permission of Fraunhofer IPA).

Manufactron is coordinated by a Super Manufactron. However, there are two major differences. Firstly, for the purpose of generating robot paths and welding parameters, the Super Manufactron issues Task Description Documents with the task type *simulation* of the Manufactrons underneath (see also Section 5.2.2). Secondly, a dedicated Simulation Manufactron is introduced which wraps a commercial robot simulation tool (see also Siemens 2013). Handling Manufactron and Simulation Manufactron share a common knowledgebase. If a robot path is requested which is not available in the knowledge base, the Super Manufactron automatically issues a simulation job to the Simulation Manufactron. The result of the simulation is automatically inserted into the knowledge base and is from now on available for execution by the Handling Manufactron.

For the automatic generation of welding parameters, the Super Manufactron issues a simulation request to the Welding Manufactron. This interfaces a commercial simulation tool and forwards the simulation request (see also Swantec 2013). After performing the simulation job, the result is stored in the Welding Manufactron's knowledgebase and is from now on available for execution.

Again, the general implementation of the Manufactrons is similar to the design illustrated in Section 5.1.3 and are communicating with the standardized Task Description Documents and Quality Result Document described in Section

5.2.2 and Section 5.2.3. Special attention is paid on the task type *simulation* described in Section “Task types of a Task Description Document” on Page 117. The Super Manufactron is implemented according to the description in Section 5.2.1 and the scheduling of the simulation and execution jobs is done by Workflow Managers according to the design illustrated in Section 5.3.1.

The major novel aspect of this setup is the usage of the same task description for both, the simulation and the execution of tasks in combination with task to method transformation for movement and welding tasks and the abstraction of those processes by a Super Manufactron. The Super Manufactron receives a task description for the production of a certain car type. The setup allows for the production of three car types (“sedan”, “station wagon” and “convertible”). Each of them requires different numbers of welding spots, spot positions and welding tasks. After the Super Manufactron received the description of the car type, it queries its knowledge base for a proper method. The method in this case is the collaboration mechanism of the Handling Manufactron and the Welding Manufactron. If such a method can be found, the Super Manufactron generates the required task descriptions for the Welding and the Handling Manufactron which then execute the respective tasks. If not method can be found, the Super Manufactron issues simulation tasks to the Manufactrons underneath.

The major benefit of this approach is that simulation and execution tasks are embedded in one environment. From the user point of view there is not difference if a task simulation or a task execution is performed. The tasks for robot movement and welding process are described once in respective Task Description Documents which are communicated to the Manufactrons using the same transport mechanism. The traditional manual operation of simulation software requires trained personnel to setup the tools. The task-driven approach allows for an easier operation as the tool complexity is covered by the Manufactrons.

On the other hand, there are a number of aspects which need to be taken into account while implementing such a setup. As mentioned already in Section 5.2.8, handling the various Task Description Documents by hand is time consuming and painful. For the three car types fifteen Task Description Documents are required: Three for the description of the overall task (make “sedan”, “station wagon” or “Convertible”) for the Super Manufactron and six for each sub Manufactron (perform welding/handling job for each car type, each with task type *simulation* and *execution*). Even if the Task Description Documents for simulation and execution have been generated automatically using the same data source (xml file), an enhanced management effort was required.

Table 13. Validation of the task-driven approach in the automotive industry.

Goal	Fulfillment	Description	Quantification
Decrease production ramp-up time	●	Embedded simulation in the Handling and the Welding Manufactron reduces the efforts for setting up the respective process.	–* (Up to 75% time reduction using virtual commissioning (Reinhart & Wünsch 2007)
Methodology for multi-variant production	◕	Knowledge on welding process execution and positioning is separated which leads to a more general use of the respective knowledge.	–*
Easy implementation in the industry	◕	Traditional robots and welding controllers have been enhanced by Manufactrons. Commercial simulation tools have been interfaced.	–*
Generate and exploit vendor's knowhow	●	Process knowhow on welding process parameters is embedded in the Welding Manufactron. By this Welding Manufacturer's vendors can provide advanced services. The Super Manufactron enables for embedding specific knowledge on the orchestration of production equipment. Hence, system integrators can deliver standardized Super Manufactrons for dedicated purposes.	Not applicable
Documentation of production data	●	Handling and Welding Manufactron consist of quality assurance systems (position of welding gun and welding spot quality). The data is gathered for the entire process.	100% quality data tracking and assessment

* Quantification available within the XPRESS Consortium, but not publishable.

Fulfillment: ● = fully achieved; ◕ = mainly achieved; ◑ = partially achieved; ○ = not achieved

Furthermore, the implementation of the Simulation Manufactron requires relatively high efforts and expertise. The respective simulation tool (in this case the Siemens tool "Process Simulate") needs to be interfaced which required proper data handling and data format transformations. The price to be paid for the task-driven interface of the Simulation Manufactron (which does not require a specific knowhow on the simulation tool) is that a knowledge system needs to be implemented within the Simulation Manufactron which automatically configures the simulation tool according to the desired simulation task (e.g. by providing proper start variables for the simulation process). In Table 13 the results of the validation of the task-driven manufacturing approach in the automotive industry are shown with focus on method generation by process simulation. It can be seen, that the result are fairly good. The task-driven approach is also feasible for planning and simulation purposes. In order to profit most of the novel approach, proper simulation tools for the processes under investigation are required. Manual interactions on the simulation and long simulation times decrease the benefit.

6.1.3 Demonstrator #3: Electrical industry

While the tests in aerospace and automotive industry focused on testing the feasibility in very large production environments, Demonstrator #3 implements a task-driven environment in electrical industry. There, quite small, but complex machines are used to produce parts of electrical components such as switches, relays, etc.

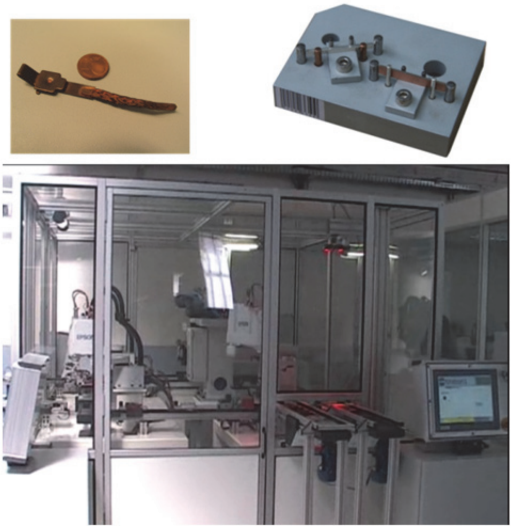


Fig. 53. Test environment in electrical applications: Usually relatively small parts are produced in electrical applications (top left). Housings are used to place and hold the products (top right). Machines with dimensions of approx. 2m x 2m consist of devices and components for transport, welding, assembly.

The applications are characterized by short production cycles and the involvement of a huge amount of different processes such as transporting, fixing, handling, welding, brazing within machines having a ground area of about four square meters. The products are often build of copper or copper-alloyed components like braids, wires or contacts and connectors.

The demonstrator is able to produce various products in random order. For demonstration purposes, products composed of studs and plates have been used which can be arranged on housings in various arrangements (see Fig. 53 upper right picture.). Various arrangements and combinations of studs and plates build the various product variants. The machine includes 4 stations which are connected by a conveyor belt. At Station 1 the housings are fed manually and are transported to Station 2. Here, the studs are inserted and after that transported to Station 3. In this station, the plates are mounted. Station 4 consists of a welding station. Herein, those products are welded, which are composed of both studs and plates. If a product consists studs only, it is bypassed to Station 4. Finally, the products are unloaded again at Station 1. The demonstrator setup is illustrated in Fig. 54:

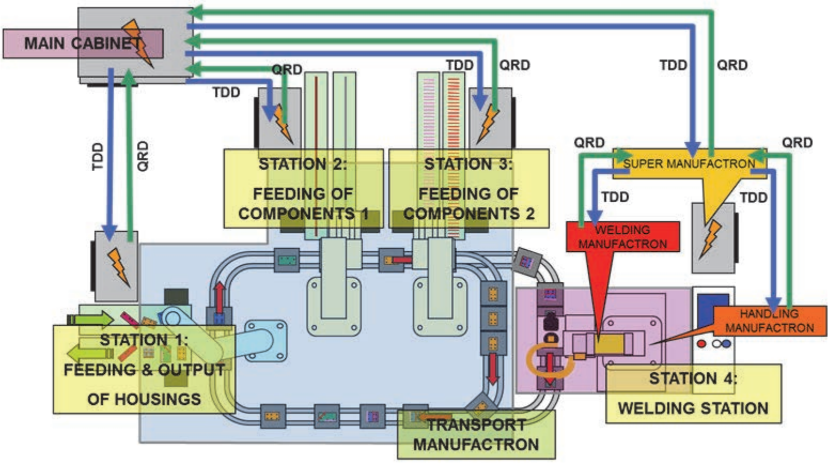


Fig. 54. Manufactronic setup of Demonstrator #3 for electrical applications (Faure 2012, published by permission of Fraunhofer IPA).

With this setup, the aim of this Demonstrator is to test the feasibility of the task-driven approach for multi-variant production. It shall demonstrate that this approach is able to reduce the change-over time for setting up the machine for a new product variant significantly. Furthermore, the capabilities on dynamic routing and overall quality data gathering shall be demonstrated.

Station 1, 2 and 3 are designed as Handling Manufactrons. For that, Station 1 consists of a SCARA robot and a visual inspection for the identification of the housing type. The Handling Manufactrons at Station 2 and 3 are composed of x/y feeders. Station 4 is composed of a welding head which is able a.) to move to various positions at the products and b.) to perform the welding process. For that, a logical separation of the welding and the positioning part of a welding head has been done by the introduction of a Handling Manufactron and a Welding Manufactron. Both Manufactrons are coordinated by a supervising Super Manufactron.

For each product variant, dedicated Task Description Documents are generated and issued by the main cabinet after the product type has been recognized at Station 1. Each Task Description Document consists of the description of the tasks to be done at the single stations with respect to the product variant. The main characteristics of the product variants are the number of studs and plates, their positions on the housing and the respective materials. According to these settings, Station 2 and Station 3 are inserting the desired studs and plates at the respective positions on the housing. Furthermore, Station 4 performs welds at the connection points of studs and plates with proper welding parameters for the materials described in the Task Description Document. Workflow Managers are responsible to guide the products from station to station and for delivering the Task Description Documents. If a product variant is composed of studs only, no welding is required. In such cases, the Workflow Manager triggers a switch in order to bypass Station 4. Quality Managers are responsible for gathering and assessing the feedback data from each Manufactron.

The novel aspect from the point of view of the machine's builder (XPRESS partner Technax) is the philosophy for the implementation of the control software. Traditionally, such machines consist of centralized control architectures in which a PLC plays the coordination role. All other devices are controlled by the PLC program in fixed program sequences. This makes multi-variant production very complicated as the control of the devices is entirely managed by the PLC program code. As a consequence, the program code gets rather complex and its adaptability to new product variants is complicated and error-prone. Furthermore, by the direct connection between PLC and device, the interface needs to be adapted whenever devices or PLC's from different vendors are used.

In comparison to those disadvantages, the task-driven approach led to a decentralized control architecture. The devices operations are triggered by the respective Manufactron after they received the incoming Task Description Documents. According to the task-driven approach, the Manufactrons consist of task to method transformation mechanisms. In the case of the Welding Manufactron, the welding task is described mainly by the material properties as explained in Section 5.1.5. For its implementation the same structure as described in Section 5.1.5 and Section 6.1.2 has been used. Modifications were required in the content of its knowledge system as the materials of the produced products differ. The Handling Manufacton's of Station 1 and 4 have not been implemented to the full extent as no flexible behavior of the handling of different housing nor of the destination positions of the welding head was required. The content of the task description of the Handling Manufactrons of Station 2 and 3 consist of the position of the studs and plates. The implementation of the respective knowledge system was relatively easy as the x/y feeders receive their commands in form of x/y/z coordinates. Therefore, only a transformation from the local coordinate system of the housing to the coordinate system of the feeders was required. As the methods have been described in text files, their modification and extension could easily been done. The demonstrator showed various advantages of the task-driven approach. Especially the aspect of multi-variant production in combination with the significant reduction of the change-over time of equipment has been highlighted.

The decentralized approach, however, led to various technical challenges and also disadvantages. Similar to the other Demonstrators, the increased complexity of the system due to its decentralized approach was challenging. The handling and maintaining of Task Description Documents and the increased number of independent entities in the system turned out to be difficult. Furthermore, interfacing the various real-time systems in order to connect them to the Manufactrons was time consuming. In addition, the general new way of organizing the machines control software requires a new way of thinking by the respective programmers. Training on the general mechanisms and challenges of distributed systems and also in programming skills (PLC programming vs. high level languages) were required. Finally, the higher costs for invest and maintenance of the required equipment is a critical issue (see also Faure 2012). The demonstrator showed various advantages of the task-driven approach. Especially the aspect of multi-variant production in combination with the significant reduction of the change-over time of equipment has been highlighted.

Table 14 summarizes the results of the validation of the task-driven manufacturing approach for a machine in electrical industry with focus on multi-variant production and decrease of change-over time. The demonstrator showed various advantages of the task-driven approach. Especially the aspect of multi-variant production in combination with the significant reduction of the change-over time of equipment has been highlighted.

Table 14. Validation of the task-driven approach for a machine in electrical industry.

Goal	Fulfill-ment	Description	Quantification
Decrease production ramp-up time	●	Welding Manufactron knowledge system vs. manual programming Feeding Manufactron knowledge system vs. manual programming	60 ➔ 30 min 60 ➔5 min
Methodology for multi-variant production	●	The centralized, PLC-based approach for the orchestration of the machine's components and the variant handling is replaced by a decentralized approach. Increase of throughput by adding a new station.	20 days ➔ 1 day
Easy imple-mentation in the industry	●	The conventional equipment has been wrapped with Manufactronic shells. Additional components like sensors, safety systems, etc. have been integrated.	No reference data available.
Generate and exploit vendor's knowhow	◕	Process knowhow on optimal welding process parameters is embedded in to Welding Manufactron. Knowledge on components positioning at the feeding stations and on visual inspection at Station 1 is also embedded in the respective Manufactrons. This allows for easy re-use of the knowledge in new machines and reduces the implementation efforts. The Super Manufactron enables for embedding of specific knowledge on the orchestration of production equipment (Station 4).	Not applicable
Documentation of production data	●	Quality data is gathered along the entire production chain of the machine.	100% data tracking

Fulfillment: ● = fully achieved; ◕ = mainly achieved; ◑ = partially achieved; ○ = not achieved

According to Faure (2012), the time for fixture changing has been decreased from one hour to some minutes and the time required to react on different production volumes decreased from several days to fifteen minutes.

6.1.4 Demonstrator #4: Photovoltaic industry

The final case study is related to the planning and the execution of a huge manufacturing scenario as well as on an overall assessment of quality data in a virtual environment. For that, a system setup typical for the manufacturing of photovoltaic wafers has been chosen. The line layout consists of eight production steps with fourteen modules (Fig. 55). Each module is represented as a Manufactron. In order to perform virtual planning and execution, a dedicated task processing time has been assigned to each Manufactron.

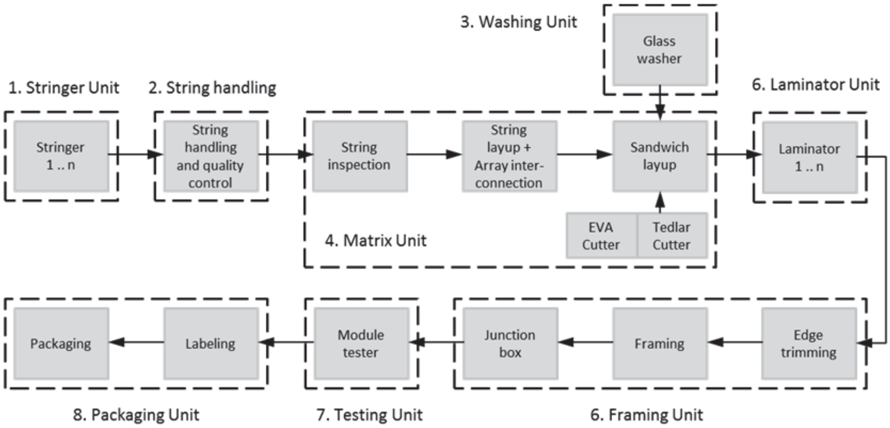


Fig. 55. Setup of the virtual environment of Demonstrator #4 in photovoltaic industry. Adapted according to Goncalves (2011a).

The Demonstrator's focus is to show the scalability of the approach and its capability for distributed simulation. Furthermore, the approach for overall quality gathering and assessment, including statistical analysis of the quality data is in focus. Simulation is done with the help of a software called *Production Simulation System* (see also Section 4.5). The Production Simulation System is capable to design line layouts, to perform distributed simulation jobs and to store simulation results in a database (Goncalves 2011b). After the design of the line layout and the definition of relevant KPI's for the simulation, simulation jobs are issued to the Manufactrons. The Manufactrons issue their simulation results in form of Quality Result Documents back to the simulation tool. An integrated optimizer provides suggestions for optimal arrangement of the Manufactrons (Ribeiro&Gonçalves 2010). In order to validate the system's capabilities on quality data gathering, a virtual execution of parallel production jobs are issued.

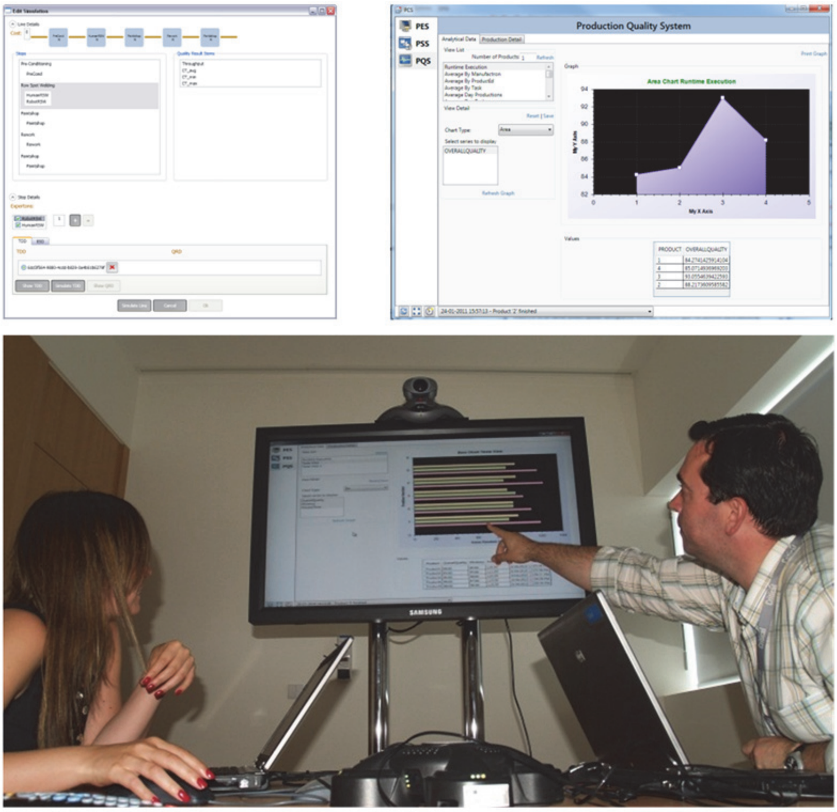


Fig. 56. Factory layout planning and quality assessment with the Production Simulation System and the Production Quality System.

For that, manufacturing jobs with the task type *emulate* are issued to the Manufactrons. The final assessment of the production quality is done by a software called *Production Quality System* (see also Section 4.5 and Fig. 56 top right). In this demonstrator, several technologies described in this thesis have been integrated. The Manufactrons have been implemented according to the specification described in Section 5.1.3. The communication and management is done via the Workflow Managers in a Workflow Execution System (Section 5.3.1). All data is communicated via Task Description Documents and Quality Result Document (Section 5.2.2 and 5.2.3). For the selection of proper (virtual) Manufactron types, the Manufactron capabilities on self-description (Section 5.2.4) have been used. Scenarios with redundant "Laminator Manufactrons" have been setup in order to test the capabilities for dynamic process routing of the

Workflow Managers in case of identified bottlenecks or insufficient quality (Section 5.3.3).

The novel elements demonstrated within this scenario is the distributed emulation and simulation of task-driven manufacturing with the help of the Workflow Managers as an unique transport mechanism. In addition, the standard way of gathering and transport of task result data with the Quality Result Documents and the Quality Managers. During the simulation phase, tasks have been issued to the several Manufactrons in parallel with the help of the Task Description Documents having the task type *simulation*. Each Manufactron issues its individual simulation result to the Production Simulation System immediately after the simulation is finalized. As soon all simulation results are available, the Production Simulation System is able to perform simulations of the entire production line using different arrangements of Manufactrons. The result of this simulation is a Workflow Manager template for the production of a certain product. The template is inserted into the Production Simulation System's knowledge system. As soon as a product needs to be produced (in the execution phase) the product description is used to query the knowledge base for a proper Workflow Manger template (see also Section 5.3.1). By this, also the novel methodology for workflow generation based on task-driven simulation could be shown.

The setup of the virtual environment could be done in short time. All Manufactrons within this scenario are based on the Manufactronic Framework and thus, only some configuration setup (e.g. Manufactron name and assignment of task emulation and execution times) were required. Major efforts were needed for the implementation of the Production Simulation System, which description is out of the focus of this thesis. The major issue identified during the tests is again related to the distributed setup of the system. In some cases, several Manufactrons did not respond on requests for simulation (e.g. due to internal malfunctions, wrong Task Description Document contents, malfunctions within the Workflow Managers and Quality Managers, network overloads). As proper standard tools for system monitoring and debugging are not available, the identification of the problems was very time consuming. As already described in Section 5.3.6, the design of the workflow system has been tailored to the needs of the partners within the XPRESS project which expressed the need of a quite restrictive dynamic behavior of the manufacturing system. For that, also the setup and the tests of this Demonstrator have been implemented within this context. Further investigation is required in order to provide authoritative conclusions on the behavior of the described approach in more sophisticated scenarios. Table 15 provides an overview of the validation results of Demonstrator #4.

Table 15. Validation results of distributed simulation, dynamic routing and quality assessment.

Goal	Fulfill-ment	Description	Quantification
Decrease production ramp-up time	●	Reduction of ramp-up time by the identification of proper Manufactrons for production tasks by means of the self-descriptive capabilities and distributed simulation in large production setups	–*
Methodology for multi-variant production	●	Multiple (virtual) products have been produced in parallel. Dynamic routing as a reaction of (simulated) quality problems or bottleneck has been tested.	–*
Easy imple-mentation in the industry		Not in focus for this Demonstrator.	
Generate and exploit vendor's knowhow		Not in focus for this Demonstrator.	
Documentation of production data	●	Quality data is gathered along the entire production chain.	100% data tracking

* Quantification available within the XPRESS Consortium, but not publishable.

Fulfillment: ● = fully achieved; ◕ = mainly achieved; ◑ = partially achieved; ○ = not achieved

6.2 Validation by related industries

This validation section reflects the task-driven approach to the characteristics of the industries related to the ones which were involved in the XPRESS project. It is based on the characterization of the industries described in Section 3.7. The final decision on the usability of the approach in industrial environments should however be based on a sufficient number of prototype tests.

6.2.1 Additive manufacturing

In additive we concentrated on direct-digital manufacturing (DDM) which allows for the production of product based directly on a computer file with no intervening tooling. The challenge for the proposed architecture is the smooth integration of DDM in a manufacturing system against the background that one-of-a-kind parts from multiple customers need to be produced and that failed parts need to be re-launched. Due to economic reasons the production of multiple products in a single build is sometimes required.

In the first appearance the task-driven approach fits well to those requirements. As the task goal is the production of the final part, a direct relation of the CAD data to the formulation of the task objective is available. Multiple quality goals (check of various geometries of one part) need to be formulated in a Task Description Document and communicated to an "*AM Manufactron*" which finally performs the job. Quality assurance systems are required for the assessment of the product quality. In case of insufficient quality feedback from the AM Manufactron, an supervising instance is required which issues a new Task Description Document having the same content. Fig. 57 illustrates a possible setup of a task-driven working cell with three AM Manufactrons and a "*Loading Manufactron*".

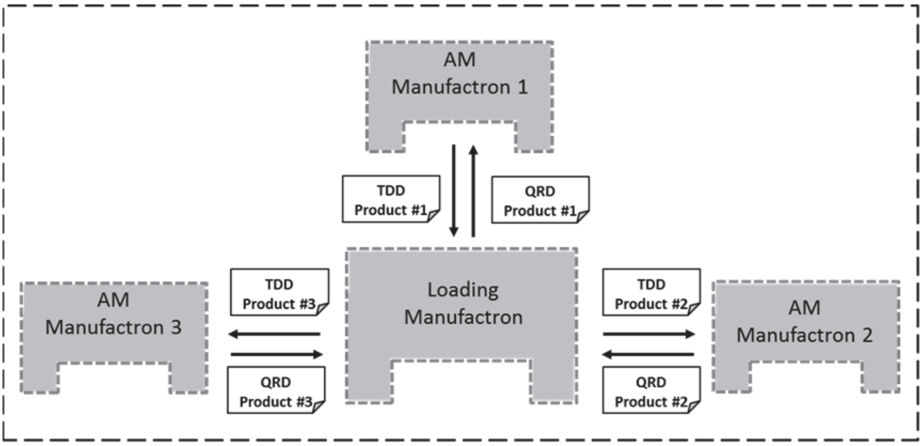


Fig. 57. Additive Manufacturing working cell with one Loading Manufactron and three AM Manufactrons.

The Loading Manufactron is a Super Manufactron and has two roles: It coordinates the AM Manufactrons and it acts as a handling unit for picking up the finished products from the AM Manufactron. In order to produce a product, the Loading Manufactron issues the respective Task Description Document to an AM Manufactron. It chooses the AM Manufactron depending on its availability. The AM Manufactron starts with the production of the product and issues a Quality Result Document when the job is finished. The Loading Manufactron picks the product, unloads it and places it on a rack outside of the working cell.

The major challenge of this scenario is the implementation of the Loading Manufactron as it needs to schedule the manufacturing jobs, to orchestrate the AM Manufactrons and to unload the parts. In particular this is challenging when the grouping of multiple parts in a single batch is required. The implementation of the AM Manufactrons should be quite easy as they receive the product's CAD data within a Task Description Document and start with the manufacturing process without any further interaction.

A potential weak point is seen in the communication scheme of the Manufactrons: As the process execution is only triggered by the arrival of Task Description Documents at the Manufactron, the Manufactron is unproductive between the finalization of one task and the start of the next task. As a consequence, the productivity of the entire manufacturing system decreases. To overcome this issue, the Manufactron need to be enabled to chain Task Description Documents in order to allow the immediate start of a new task as soon as the Manufactron completed the task before. Furthermore, as the production of a part is provided as one large task, the AM Manufactrons are "blocked" during operation. They are thus inflexible and unable to react on disturbances properly.

6.2.2 Heat treatment

The challenge for a manufacturing control system in heat treatment is to optimize the equipment utilization and the lead time of production. For that, parts coming from different orders having different geometry but require similar heat treating need to be grouped which can then brought together in the respective equipment. In a task-driven environment, a possible setup is shown in Fig. 58. The scheduling of products with respect to their requirements on heat treatment is in the knowledge domain of a "*Heating Manufactron*". As the scheduling activity has to be done in advance of the heating process, a "simulation" is done. This means that the Heating Manufactron builds batches of products having the same heat treatment requirements based on the product properties taking also the envisaged optimization goal (e.g. equipment utilization or lead time reduction) into account. According to the optimal constellation, respective Task Description Documents will be created (Fig. 58 left).

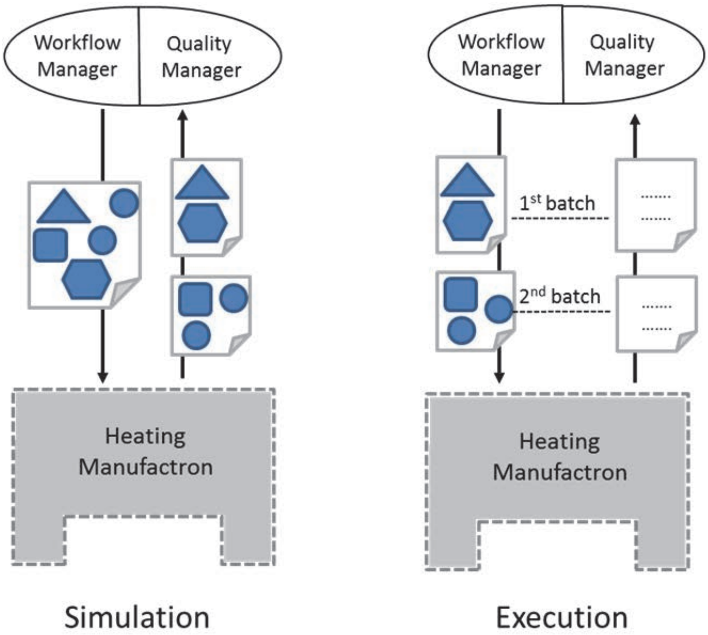


Fig. 58. Heating Manufactron used in simulation mode for part grouping and in execution mode for heat treatment of similar parts.

In the execution phase, a Workflow Manager instance is created which communicates the Task Description Documents to the Heating Manufactron. After the heat treatment, the Heating Manufactron issues Quality Result Documents to the Quality Manager (Fig. 58 right).

The challenge in this scenario is the implementation of the Heating Manufactron with scheduling of the orders and the dynamic creation of Task Description Documents. For that, proper task-to-method transformations need to be implemented which choose and/or create heat treatment profiles on the basis of the properties of the parts. Furthermore, the scheduling has to be implemented in order to arrange the parts with respect to their heating requirements and the envisaged optimization goal.

6.2.3 Sheet metal industries

According to its classification, the challenge in sheet metal processing is the 2D nesting of different parts on the same metal sheet, followed by bending and eventually assembly. In each process step the grouping of the parts varies.

Dynamic nesting requires periodic dynamic grouping of parts with the same base material from different orders. This is –similar to the grouping of parts in heat treatment – a scheduling task. For the further validation of the task-driven approach in sheet metal industry we therefore concentrate on (static) nesting of parts only.

Against this context, the following process flow is taken as an example for the further discussion: In the first station, the punching is done. A number of different products shall be produced at this station which requires one or multiple similar metal sheets. The challenges in this station are a) to decide on the grouping of the different parts on the metal sheet(s) in such way that the optimization goal (e.g. lead time, minimum waste) is reached and b) to perform the punching process. After that, in the second station, bending is done. To do so, similar parts need to be grouped and the proper setup of the bending machine needs to be done. In this station, multiple similar products can be bended simultaneously. In the last station, multiple products are assembled together.

In a task-driven environment the setup illustrated in Fig. 59 could be used. In advance of the execution, a ”*Grouping Manufactron*” is used which decides on the grouping of the parts on the metal sheet(s).

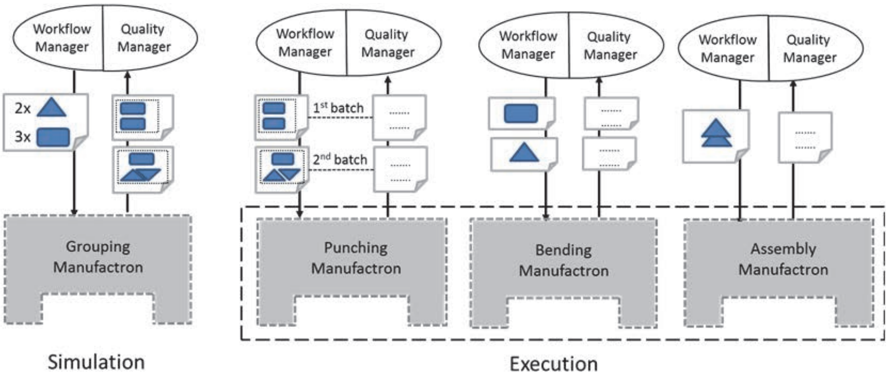


Fig. 59. Sheet metal processes with a Grouping Manufactron for building the product batches in simulation phase and Punching, Bending and Assembly Manufactron in execution phase.

The input for the Grouping Manufactron is the product data, especially the geometry of the products expressed in a Task Description Document. The Grouping Manufactron ”simulates” the grouping with respect to the desired optimization goal (lead time, minimum waste, etc.). The result of the simulation process is the arrangement of the parts on one or multiple metal sheets (Fig. 59 left). For production, a Task Description Document is generated in which the sub tasks for part punching are arranged according to the simulation result. Furthermore, also the grouping of similar tasks for bending and the grouping of different parts for assembly needs to be described. For the transportation of the Task Description Documents Workflow Managers are created.

For the example scenario, task-driven manufacturing fits, but significant handling of Task Description Documents is required (e.g. distinguished Task Description Documents for simulation and execution tasks require copy&paste work for sub tasks, geometry data of the parts, etc.). To setup the task-driven environment, also the respective Manufactrons need to be implemented. Proper task-to-method-transformation in order to realize the arrangement task of the Grouping Manufactron as proposed as well as the setup of the punching, bending and assembly processes. In general, the introduction of the task-driven approach for sheet metal industry is more challenging as for the industries described above.

6.3 Discussion

The task-driven approach was validated in various application scenarios. Besides the tests of the planning and simulation capabilities described in Section 6.1.4 and the validation of industries which were not involved in the XPRESS project, all tests were carried out in realistic industrial setups. Moreover, the demonstration setup in the aerospace industry was even integrated into a running production system. The following conclusions can be drawn with respect to the three core elements of the task-driven approach:

The feasibility of the approach to *task-driven devices at shop-floor* level proposed in Section 5.1 was demonstrated in all test cases. All Manufactrons were implemented according to the proposed specifications. Most of them also implemented sophisticated algorithms for task interpretation and task to method mapping according to the mechanisms described herein. Potential improvements were identified in terms of reliability of the algorithms for the task to method mapping in case of incomplete or contradictory data in the knowledgebase.

The *hierarchical organization and communication model* described in Section 5.2.2 was also tested in all demonstration scenarios. The feasibility of delivering task descriptions with standardized Task Description Documents to the Manufactrons and for gathering quality data with Quality Result Documents from the Manufactrons were shown. The test revealed that the communication of Task Description Documents through the various levels can lead to high traffic volumes throughout the entire network and to loss in performance in the Manufactrons despite of high data transmission. Hence, sufficient network capacity and efficient processing algorithms are required. In addition, the feasibility of a hierarchical system organization at shop-floor level by Super Manufactrons was demonstrated. As expected, the hierarchical structure allows for the reduction of complexity by the aggregation of knowledge. On the other hand, a malfunction within the Super Manufactron's host computer leads to a breakdown of the related Super Manufactron and its related Sub-Manufactrons. For this reason, proper strategies for disturbance handling are required. In addition, a more flexible orchestration of production equipment underneath the Super Manufactrons could be useful. Super Manufactrons should be equipped with capabilities for dynamic allocation of resources, e.g. in case of unexpected production volumes or if the performance of a Sub-Manufactron decreases.

The concepts described in Section 5.3.1 were used for the *orchestration of production equipment* by Workflow Managers and quality data gathering as well as the dynamic routing. It became evident that the approaches are useful within large production environments as well in smaller production setup or even within a single machine. In order to improve the efficiency of the system, the workflow generation can be supported by the dynamic creation of Task Description Documents out of the self-descriptive capabilities of the Manufactrons. Furthermore, the next evolution of the Workflow Managers could be "Product Managers" which represent and encapsulate knowledge on the products. This would enable a more product-driven production and would help to reduce the complexity and centralization of the Production Configuration System at ERP level.

The validation of the proposed architecture with respect to the related industries showed that it can also be adapted to other industries. The efforts for such an adaptation are mainly related to the target of the respective setup: If the setup places a stronger focus on the processes at shop floor level with a relatively fixed routing of products, the major challenge is to implement the respective Manufactrons. Here, mainly the knowledge system including the formalization of process knowhow and the task-to-method transformation as well as proper quality assessment systems need to be developed. As those features are related to the behavior of the respective process, extensive development efforts are expected even if generic mechanisms are available as described in Pollak *et al.* (2011). If the focus of the setup is more on the MES level including dynamic routing and order management, further adaptation efforts are expected. As order management and scheduling were not key issues of the XPRESS project, only little effort was put into this area. In these types of setups, dedicated Manufactrons for order and product management might be introduced in order to obtain clear responsibilities assigned to the Manufactrons for the various roles in a manufacturing system.

7 Summary and conclusions

This thesis aims to develop an architecture for flexible manufacturing systems which allows for easy take-up in the industry with a special focus on the traditional industry sectors. On the one hand, they need to introduce flexible and extensible manufacturing systems. However, on the other hand, they also have to face restrictions and boundary conditions which hamper the implementation of these new solutions. Therefore, an architectural design is required which fits their needs.. Such an architecture is supposed to fulfill five main goals:

- Decrease production ramp-up time;
- Provide an easy-to-use methodology for multi-variant production;
- Allow for an easy implementation in the industry;
- Generate and use knowhow of component and system vendors;
- Comprehensive documentation on production data.

The current state of the art in manufacturing systems is described in Chapter 2. It places a strong focus on the readiness of their implementation in the industry. Chapter 3 describes the requirement gathering process within the three representative branches aerospace, automotive and electrical industry which was part of the European project XPRESS. In total, ten requirements were identified. Chapter 4 describes the system design and architecture taking into account the requirements and the most important scientific approaches. The core elements of the novel approach are intelligent production devices which receive only a description of a task to be performed and are able to map the task description to optimal process execution methods. Devices equipped with such features are defined as Manufactrons. Chapter 5 describes the realization of such a manufacturing system. To do so, three main pillars of the system architecture are described in detail in dedicated sub-sections:

- Task-driven manufacturing;
- Hierarchical organization and communication;
- Quality-oriented workflow.

At the end of each sub-section, a reflection of the architecture features including the related requirements is given. Finally, in Chapter 6 four industrial demonstration scenarios are described in which the task-driven approach was tested. The performance of each Demonstrator is measured with respect to the five main goals of this thesis.

The general performance of the system design is very good for all Demonstrators. The validation of the Demonstrators with respect to the thesis' objectives shows that the objectives are either largely or even completely achieved. A highlight of the demonstration is the integration of major parts of the architecture in a real production environment of the aerospace industry as shown in Demonstrator #1. Here, the feasibility of the approach and the coexistence of traditional and task-driven manufacturing paradigm are shown. The conclusion of the Demonstrator #2 implementation is also positive as the feasibility of the general communication mechanisms of Task Description Documents and Quality Result Documents were also proven for the communication of emulation and simulation jobs. The separation of domain knowledge by the implementation of dedicated Manufactrons for welding, handling and simulation reduces the complexity of the system and enables exploiting specific knowledge of the respective vendors. However, on the other hand, there is a lot of work required for communication and synchronization. With Demonstrator #3, the successful implementation of the architecture, including smaller manufacturing setups, was shown. This demonstrates the large bandwidth which can be covered by the proposed architecture. From a technical point of view, the former completely centralized PLC-based machine design was replaced by a decentralized architecture, attaining the objectives of a rapid introduction of new product variants, a fast change-over of the complete setup and the easy integration and exchange of equipment. Demonstrator #4 then showed the feasibility of the architecture for large system setups, including distributed simulation and emulation as well as the (virtual) production of a huge amount of products.

With these results, the general feasibility of the proposed architecture was successfully demonstrated. Even though the requirements were compiled by representatives of only three industry sectors, the requirements are not very specific or even tailored to these industries. The validation of the architecture for the three related industries showed that most of the demands also apply to other traditional industries. That is the reason why the proposed flexible architecture based on task-driven agents provides great potential for other sectors as well.

This research work moves the field forward and advances the state of the art. The main contribution is the introduction of *task-driven process execution on shop floor*. With the Manufactrons, agile agents are available which encapsulate the knowledge for a specific domain. This contributes to the transition from a resource-based to a knowledge-based production paradigm which is seen as one of the success factors for future manufacturing systems.

Another contribution is the *integration of physical devices* by a Manufactronic shell. This contributes to a better consideration of equipment and controls in multi-agent systems which is seen as one weak point of existing architectures. This approach allows for interfacing existing traditional devices, which can reduce the barriers for integrating multi-agent systems into industrial environments.

The Super Manufactrons allow for *abstraction in a task-driven environment*. Similar to the ADACOR supervisor holons, Super Manufactrons can also assist optimization in distributed environments. As an extension, Super Manufactron can receive abstract task descriptions and can have different roles at the same time: A coordination role for underlying sub Manufactrons and an executive role for controlling physical devices.

The success of the task-driven architecture in *industrial implementation* can also contribute to a better market penetration of multi-agent systems as it showed the advantages of such a system in different implementations of the architecture. Its integration in a real-life production environment also showed the coexistence of traditional setups and task-driven, agent-based technologies.

With the introduction of the Task Description Documents and the Quality Result Documents, *an approach for the description of task and quality information* is available which can be used for the formulation of simulation, emulation and execution of tasks.

For *quality data tracking* along the production line, Quality Managers were introduced. In combination with the Workflow Managers this provides a basis for product routing with respect to the quality results of processes.

References

Almeida F, Dias P, Gonçalves G, Peschl M & Hoffmeister M (2011) A proposition of a manufactronic network approach for intelligent and flexible manufacturing systems. IJIEC 2(4): 873–890.

Almeida F, Terra B, Dias P & Gonçalves G (2010) Transport with automatic guided vehicles in the factory of the future. Proc IEEE Conference on Emerging Technologies and Factory Automation (ETFA): 1–4.

Angelsmark O, Malec J, Nilsson K, Nowaczyk S & Prosperi L (2007) Knowledge Representation for Reconfigurable Automation Systems. Proc International Conference on Robotics and Automation (ICRA-07). Rome, Italy, 10 April.

Babiceanu RF (2005) Holonic-based Control System for Automated Material Handling Systems. Faculty of the Virginia Polytechnic Institute and State University. Blacksburg, Virginia.

Baltec (2013) Process Controller HPP-25, the High Performance Package. URI: http://www.baltec.com/en/products/process-monitoring/hpp-25.html, checked on 21/09/2013.

Bauer A, Browne J, Bowden R & Duggan J (1994) Shop floor control systems. From design to implementation. 2nd ed. London, New York: Chapman & Hall.

Bengel M (2008) Modelling objects for skill-based reconfigurable machines. In: Pham DT, Eldukhri EE & Soroka AJ (eds) Innovative production machines and systems. Third I*PROMS Virtual Conference, 2nd-13th July 2007, 3/2007. Dunbeath Scotland, Boca Raton FL, Whittles, CRC Press: 238–243.

Black JT, DeGarmo EP & Kohser RA (2013) DeGarmo's materials and processes in manufacturing. 11th ed. Upper Saddle River NJ, John Wiley & Sons.

Blanc P, Demongodin I & Castagna P (2008) A holonic approach for manufacturing execution system design: An industrial application. Engineering Applications of Artificial Intelligence 21(3): 315–330.

Booch G (2007) Object-oriented analysis and design with applications. 3rd ed. Upper Saddle River NJ, Addison-Wesley.

Botti V & Giret A (2008) ANEMONA. A multi-agent methodology for holonic manufacturing systems. London, Springer.

Browne J (1988) Production activity control—a key aspect of production control. International Journal of Production Research 26(3): 415–427.

Brückner S, Wyns J, Peeters P & Kollingbaum M (1998) Designing Agents for Manufacturing Control. In: Interrante LD & Luger GF (eds) Proceedings of the 1998 Artificial Intelligence and Manufacturing Research Planning Workshop: AAAI Press Albuquerque New Mexico: 40–46.

Bussmann S & Schild K (2001) An Agent-Based Approach to the Control of Flexible Production Systems. Proc. 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETFA 2001) Vol. 2: 481–488.

Cândido G & Barata J (2007) A Multiagent Control System for Shop Floor Assembly. In: Mařík V, Vyatkin V & Colombo AW (eds) Holonic and Multi-Agent Systems for Manufacturing vol. 4659. Berlin Heidelberg, Springer: 293–302.

Castellini P, Cristalli C, Foehr M, Leitao P, Paone N, Schjolberg I, Tjonnas J, Turrin C & Wagner T (2011) Towards the integration of process and quality control using multi-agent technology. Proc IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society: 421–426.

Chao C & Helal A (2009) Device Integration in SODA Using the Device Description Language. Proc Applications and the Internet, 2009. SAINT '09: 100–106.

Chen FF & Adam EE (1991) The impact of flexible manufacturing systems on productivity and quality. IEEE Trans Eng Manage 38(1): 33–45.

Christensen JH (1994) Holonic Manufacturing Systems: Initial Architecture and Standards Directions. Proc of First European Conference on Holonic Manufacturing Systems, Hannover.

Chryssolouris G (2006) Manufacturing Systems: Theory and Practice. New York NY, Springer.

Colombo AW, Schoop R & Neubert R (2005) An agent-based intelligent control platform for industrial holonic manufacturing systems. IEEE Transactions on Industrial Electronics 53(1): 322–337.

Demig (2004) Quality and Process Optimization of the Heat Treatment. Demig Prozessautomatisierung GmbH. URI: http://demig.de/cms/upload/download/Beherschte_Diffusion_13.07.05_Englisch.pdf, checked on 29/09/2013.

Dilts DM, Boyd NP & Whorms HH (1991) The evolution of control architectures for automated manufacturing systems. Journal of Manufacturing Systems 10(1): 79–93.

Drath R, Luder A, Peschke J & Hundt L (2008) AutomationML - the glue for seamless automation engineering. IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) 2008: 616–623.

Duffie NA & Prabhu VV (1994) Real-time distributed scheduling of heterarchical manufacturing systems. Journal of Manufacturing Systems 13(2): 94–107.

EDDL (2013) Electronic Device Description Language (EDDL). URI: http://www.eddl.org/Pages/default.aspx, checked on 22/09/2013.

Eickhorst D & Trostmann E (2012) Manufactronic concepts for fuselage shell assembly. In: Hoffmeister M & Peschl M (eds) Knowledge-based manufacturing. Stuttgart, Fraunhofer: 84–90.

Eigenbrod H (2008) SIARAS. Publishable Results. URI: http://cordis.europa.eu/documents/documentlibrary/121979181EN6.pdf, checked on 21/07/2013.

EC (2011) FoF.NMP.2012-3. Intelligent production machines and “plug and-produce” devices for the adaptive system integration of automation equipment, robots and other intelligent machines, peripheral devices, smart sensors and industrial IT systems. URI: http://ec.europa.eu/research/participants/portal/page/cooperation?callIdentifier=FP7-2012-NMP-ICT-FoF#wlp_call_FP7, checked on 4/10/2013.

Faure D (2012) Highly flexible and multi-variant production in electrical industry. In: Hoffmeister M & Peschl M (eds) Knowledge-based manufacturing. Stuttgart, Fraunhofer: 106–111.

Fedrowitz C (2012) Planning process and automatic robot path generation in automotive industry. In: Hoffmeister M & Peschl M (eds) Knowledge-based manufacturing. Stuttgart, Fraunhofer: 91–96.

Feng Q, Bratukhin A, Treytl A & Sauter T (2007) A Flexible Multi-Agent System Architecture for Plant Automation. Proc 5th IEEE International Conference on Industrial Informatics 2007 vol 2: 1047–1052.

Finin T, Fritzson R, McKay D & McEntire R (1994) KQML as an agent communication language. Proc Third international conference on Information and knowledge management. New York: 456–463.

FIPA (2002a) FIPA ACL Message Structure Specification (SC00061G). URI: http://www.fipa.org/specs/fipa00061/, checked on 13/09/2013.

FIPA (2002b) FIPA Agent Communication Language Specifications. URI: http://www.fipa.org/repository/aclspecs.html, checked on 13/09/2013.

FIPA (2002c) FIPA Device Ontology Specification (SC00091E). URI: http://www.fipa.org/specs/fipa00091/index.html, checked on 13/09/2013.

FIPA (2003) Publicly Available Agent Platform Implementations. URI: http://fipa.org/resources/livesystems.html, checked on 13/09/2013.

Fletcher M, McFarlane D, Thorne A, Jarvis D & Lucas A (2003) Evaluating a Holonic Packing Cell. In: Mařík V, McFarlane D & Valckenaers P (eds) Holonic and Multi-Agent Systems for Manufacturing vol 2744: Berlin Heidelberg, Springer: 246–257.

Gerber C, Hanisch HM & Ebbinghaus S (2008) From IEC 61131 to IEC 61499 for distributed systems: a case study. EURASIP Journal of Embedded Systems 4(1–4): 8.

Gigler P, Hoffmeister M & Peschl M (2012) Workshop 2. Workflows in precision assembly production. In: Hoffmeister M & Peschl M (eds) Knowledge-based manufacturing. Stuttgart, Fraunhofer: 35–49.

Goncalves J (2011a) XPRESS Project: PCS demonstration documentation. Deliverable D-DEMO 2. Project XPRESS: IP 026674-2 (Ed) Porto, Portugal.

Goncalves J (2011b) XPRESS Project: Final implementation of the PCS. Deliverable D-RTD 12.2. Project XPRESS: IP 026674-2 (Ed) Porto, Portugal.

GRACE (2013) Grace Project. URI: http://grace-project.org/, checked on 21/07/2013.

Grasse PP (1959) La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. La theorie de la stigmergie: essai d'interpretation du comportement des termites constructeurs. In Insectes Sociaux 6: 41–81.

Griffin C (2009) Knowledge-based machining with DS PLM. Dassault Systèmes. URI: http://www.3ds.com checked on 7/09/2012.

Hartmann B, Link N & Trommer GF (2010) Indoor 3D position estimation using low-cost inertial sensors and marker-based video-tracking. Position Location and Navigation Symposium (PLANS): 319–326.

Hartonas T (2006) The ProsAgent Framework. Integrating Holonic and Multi-Agent based Approaches to Industrial Process Control. Annals of Mathematics, Computing & Teleinformatics 4: 40–50.

Hatvany J (1985) Intelligence and cooperation in heterarchic manufacturing systems. Robotics and Computer-Integrated Manufacturing 2: 101–104.

Herrmann JW & Delalio DR (2001) Algorithms for sheet metal nesting. IEEE Transactions on Robotics and Automation 17(2): 183–190.

Hildebrand T (2005) Theoretische Grundlagen der bausteinbasierten, technischen Gestaltung wandlungsfähiger Fabrikstrukturen nach dem PLUG+PRODUCE Prinzip: Techn. Universität Chemnitz, Institut für Betriebswissenschaften und Fabriksysteme.

Hoffmeister M (2008) XPRESS Project: Requirement list for each component. Deliverable D-RTD 1.1. Project XPRESS: IP 026674-2 (Ed) Karlsruhe.

Hoffmeister M (2012) Modelle zur aufgabengeführten Produktionsausführung in der wandlungsfähigen Produktion. Universität Stuttgart, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen.

Hoffmeister M, Peschl M, Wertz R & Verl A (2011) Task Description Documents - An Interface Standard For Factory Automation. International Journal of Industrial Engineering Theory, Applications and Practice (IJIE 2011). Stuttgart: 222–228.

Hoffmeister M & Peschl M (eds) (2012) Knowledge-based manufacturing. A new approach for precision assembly industries. Status colloquium, European project XPRESS. Status Colloquium "Knowledge-based Manufacturing". Stuttgart, 03.-04.05.2011. Stuttgart, Fraunhofer IPA.

HWH (2009) Harms & Wende Schweißzeit 02/2009. URI: http://www.harms-wende.de, checked on 11/02/2013.

Idea (2013) Automatisch rekonfiguriert - Die flexible Fertigungszelle SIARAS. Edited by Idea TV GmbH. URI: http://www.innovations-report.de/html/berichte/messenachrichten/bericht-111666.html, checked on 21/07/2013.

IDEAS (2013) The IDEAS Project. Edited by IDEAS Consortium. URI: http://www.ideas-project.eu, checked on 21/07/2013.

IEC (2003a) IEC61331-1, Part 1: General information.

IEC (2003b) IEC61804-1, Function blocks for process control - Part 1: Overview of system aspects. IEC.

IEC (2005) IEC61499-1, Function Blocks - Part 1 Architecture.

IEC (2010a) IEC61158, Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series.

IEC (2010b) IEC 61784, Industrial communication networks - Profiles - Part 1: Fieldbus profiles. IEC.

IEC (2010c) IEC61804-3, Function blocks for process control - Part 3: Electronic Device Description Language (EDDL). IEC.

IEC (2013) IEC61131-9, Function Blocks - Part 9 Single-drop digital communication interface for small sensors and actuators (SDCI).

IMS (2012) Intelligent Manufacturing Systems. Global Research and Business Innovation Program. URI: http://www.ims.org/, checked on 29/08/2012.

JACK (2013) JACK® Key Attributes. URI: http://www.agent-software.com/products/jack/, checked on 2/08/2013.

Jennings NR (2001) An agent-based approach for building complex software systems. Communications of the ACM 44(4): 35–41.

JSON (2011) JSON: The Fat-Free Alternative to XML. URI: http://www.json.org/fatfree.html, checked on 21/09/2013.

Khamt N, Murata M, Kuboki T, Shibata T & Jin Y (2012) Effects of Punch and Die Parameter on Spring-back at Bending Process of Thick Sheet A5083. Journal of the Japan Society for Technology of Plasticity 53(615): 362–366.

Koestler A (1989) The ghost in the machine. London, Arkana.

Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G & van Brussel H (1999) Reconfigurable Manufacturing Systems. CIRP Annals - Manufacturing Technology 48(2): 527–540.

Koskimäki H, Huikari V, Siirtola P & Röning J (2013) Behavior modeling in industrial assembly lines using a wrist-worn inertial measurement unit. Journal of Ambient Intelligence and Humanized Computing 4(2): 187–194.

KUKA (2013) KUKA Industrial Robots. URI: http://www.kuka-robotics.com/en/, checked on 3/10/2013.

Lastra JLM, Insaurralde C & Colombo A (2009) Agent-based Control for Desktop Assembly Factories. In: Wang L & Nee AYC (eds) Collaborative Design and Planning for Digital Manufacturing. London, Springer: 265–291.

Leitão P (2004) An Agile and Adaptive Holonic Architecture for Manufacturing Control. University Porto, Faculty of Engineering.

Leitão P (2009) Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence 22(7): 979–991.

Leitão P & Restivo F (2003) Identification of ADACOR holons for manufacturing control. In: Monostori L, Kádár B & Morel G (eds) Intelligent manufacturing systems 2003. (IMS 2003): A proceedings volume from the 7th IFAC workshop, Budapest, Hungary, 6–8 April 2003: 109–114.

Leitão P & Restivo F (2006) ADACOR: A holonic architecture for agile and adaptive manufacturing control. Computers in Industry 57(2): 121–130.

Lind M, Roulet-Dubonnet O, Nyen P, Gellein LT, Lien T & Skavhaug A (2009) Holonic Manufacturing Paint Shop. In: Mařík V, Strasser T & Zoitl A (eds) Holonic and Multi-Agent Systems for Manufacturing vol 5696. Berlin Heidelberg, Springer: 203–214.

Lüder A, Peschke J, Bratukhin A, Treytl A, Kalogeras A & Gialelis J (2006) The PABADIS'PROMISE - architecture. Proc ANIPLA 2006 International Congress "Methodologies for Emerging Technologies in Automation", Rome, Italy, Nov. 2006.

Maier RRJ, MacPherson WN, Barton JS, Carne M, Swan M, Sharma JN, Futter SK, Knox DA, Jones BJS & McCulloch S (2013) Embedded Fiber Optic Sensors Within Additive Layer Manufactured Components. IEEE Sensors Journal 13(3): 969–979.

MANUFUTURE (2004) MANUfuture - a vision for 2020. Assuring the future of manufacturing in Europe. Directorate-General for Research, European Commission (ed) Office for Official Publications of the European Communities, Luxembourg.

Martensson P, Gullander P & Klingstam P (1997) Generic modelling of cell control systems and production resources. Proc 7th Conference on Flexible Automation and Intelligent Manufacturing, FAIM ´97, 1997: 123–134.

MASCADA (1997) Mascada home page. URI: http://www.mech.kuleuven.be/mascada/, checked on 9/09/2012.

Matuschek (2012) High Current Resistance Welding Solutions. URI: http://www.matuschekwelding.com, checked on 11/02/2013.

Mendikoa I, Sorli M, Armijo A, Garcia L, Erausquin L, Insunza M, Bilbao J, Friden H, Björk A, Bergfors L, Skema R, Alzbutas R & Iesmantas T (2013) Energy Efficiency Optimisation in Heat Treatment Process Design. In: Emmanouilidis C, Taisch M & Kiritsis D (eds) Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services vol 397. Berlin Heidelberg, Springer: 127–134.

Miller (2012) Handbook for resistance spot welding. Miller Electric MFg Co. URI: http://www.millerwelds.com/pdf/Resistance.pdf, checked on 25/09/2012.

Mönch L, Stehli M, Zimmermann J & Habenicht I (2006) The FABMAS multi-agent-system prototype for production control of water fabs: design, implementation and performance assessment. Production Planning and Control 17(7): 701–716.

Mönch L & Stehli M (2006) ManufAg: a multi-agent-system framework for production control of complex manufacturing systems. Information Systems and e-Business Management 4(2): 159–185.

Monostori L, Váncza J & Kumara SRT (2006) Agent-based systems for manufacturing. In: The International Academy for Production Engineering CIRP (ed) CIRP Annals-Manufacturing Technology 55. 2nd ed: 697–720.

Mühe H, Angerer A, Hoffmann A & Reif W (2010) On reverse-engineering the KUKA Robot Language. 1st International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob'10), 2010. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010).

Murrell P (2007) XML: Advantages and disadvantages. Vienna University of Economics and Business: Institute for Statistics and Mathematics. URI: http://statmath.wu.ac.at/courses/data-analysis/itdtHTML/node57.html, checked on 21/09/2013.

Nurseitov N, Paulson M, Reynolds R & Izurieta C (2009) Comparison of JSON and XML Data Interchange Formats: A Case Study. CAINE'09: 157–162.

Ollikainen M & Varis J (2005) Analysis of bending processes of sheet metal parts and improving them with new investments - a Finnish case study. Mechanika (Kaunas) (4): 65.

Onori M, Lohse N, Barata J & Hanisch, C (2012) The IDEAS project: plug & produce at shop-floor level. Assembly Automation 32(2): 124–134.

Oracle (2011) Recipe & Material Workspace Recipe Management Guide. URI: http://docs.oracle.com/cd/E18601_08/otn/pdf/user/E18632_02.pdf, checked on 5/12/2011.

PABADIS-PROMISE (2008) Industrial Application of the PABADIS'PROMISE System. White Paper. CVS at IAF, Univ of Magdeburg. URI: http://www.uni-magdeburg.de/iaf/cvs/pabadispromise/dokumente/whitepaper2_v60.pdf, checked on 27/07/2013.

Peschke J, Luder A & Kuhnle H (2005) The PABADIS'PROMISE architecture - a new approach for flexible manufacturing systems. Proc 10th IEEE Conference on Emerging Technologies and Factory Automation, 2005. ETFA 2005: 490–496.

Peschl M (2008a) XPRESS Project: Final concept documentation. Deliverable D-RTD 2.4. Project XPRESS: IP 026674-2 (Ed) with assistance of Link N & Hoffmeister M. Karlsruhe

Peschl M (2008b) XPRESS Project: Checklist for each process. Deliverable D-RTD 2.1. Project XPRESS: IP 026674-2 (Ed) Karlsruhe.

Peschl M (2010) XPRESS Project: Demonstrator Roadmap. Report on Demonstrators, internal report. Project XPRESS: IP 026674-2 (Ed) Karlsruhe.

Peschl M (2011) XPRESS Project: Report on Manufactron guided Production flow, Manufactronic Machines and Human Integration demonstration. Deliverable D-DEMO 1. Project XPRESS: IP 026674-2 (Ed) Karlsruhe.

Peschl M, Link N, Hoffmeister M, Gonçalves G & Almeida F (2011) Designing and implementation of an intelligent manufacturing system. JIEM 4(4): 718–745.

Peschl M, Röning J & Link N (2012a) Distributed simulation for task-driven flexible manufacturing systems. In: Katalinić B (ed) 23rd DAAAM International symposium Conference papers. Vienna: DAAAM International: 85–88.

Peschl M, Röning J & Link N (2012b) Human integration in task-driven flexible manufacturing systems. In: Katalinić B (ed) 23rd DAAAM International symposium Conference papers. Vienna: DAAAM International: 171–174.

Peschl M & Hoffmeister M (2011) A task-driven flexible manufacturing system for major industrial applications. In: Katalinić B (ed) Annals & Proceedings of the 22nd DAAAM World Symposiums. Vienna, Austria, 23–26 November 2011: 1463–1464.

Preusse S (2011) XPRESS Project: Validation results document. Deliverable D-INNO 3.6. Project XPRESS: IP 026674-2 (Ed) Karlsruhe.

PRIME (2013) PRIME project webpage. Edited by PRIME Consortium. URI: http://www.prime-eu.com./, checked on 21/07/2013.

QS-Technologies (2013) INLINE Prozessüberwachung. URI: http://www.hwh-qst.de/index.php/home-en.html, checked on 15/09/2013.

Reinhart G & Wünsch G (2007) Economic application of virtual commissioning to mechatronic production systems. Production Engineering 1(4): 371–379.

Pollak J, Sarveniazi A & Link N (2011) Retrieval of process methods from task descriptions and generalized data representations. The International Journal of Advanced Manufacturing Technology 53: 829–840.

Ribeiro L, Barata J, Onori M, Hanisch C, Hoos J & Rosa R (2011) Self-organization in automation - the IDEAS pre-demonstrator. Proc IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society: 2752–2757.

Ribeiro L, Barata J, Cândido G & Onori M (2010) Evolvable Production Systems: An Integrated View on Recent Developments. Proc 6th CIRP-Sponsored International Conference on Digital Enterprise Technology vol 66. Berlin Heidelberg, Springer: 841–854.

Ribeiro T & Gonçalves G (2010) Formal methods for reconfigurable assembly systems. Proc IEEE Conference on Emerging Technologies and Factory Automation (ETFA): 1–6.

Rukki (2009) Resistance Welding Manual. Rautaruukki Corporation. URI: http://www.ruukki.com/~/media/Files/Steel-products/Ruukki-Resistance-welding-manual.pdf, checked on 25/09/2012.

Russell SJ, Norvig P & Davis E (2010) Artificial intelligence. A modern approach. 3rd ed. Upper Saddle River, Prentice Hall.

Scribner K (2007) Microsoft Windows Workflow Foundation step by step. Redmond WA, Microsoft Press.

Senehi MK, Kramer TR & Ray SR (1994) Hierarchical control architectures from shop floor level to end effectors. In: Joshi SB & Smith JS (eds) Computer Control of Flexible Manufacturing Systems: Research and Development. Chapman & Hall: 31–60.

Shipp SS, Gupta N, Lal B, Scott JA, Weber CL, Finnin MS, Blake M, Newsome S & Thomas S (2012) Emerging Global Trends in Advanced Manufacturing. Institute for defense analyses (Ed).

Siemens (2013) Process Simulate. Edited by SIEMENS AG. URI: http://www.plm.automation.siemens.com/en_us/products/tecnomatix/assembly_planning/process_simulate/index.shtml, checked on 29/12/2012.

Siemens (2009) Robcad: Siemens PLM Software. URI: http://www.plm.automation.siemens.com/de_de/products/tecnomatix/robotics_automation/robcad, updated on 31/07/2009, checked on 15/09/2013.

Skill-Pro (2013) Skill Pro project website. Edited by Skill-Pro Consortium. URI: http://www.skillpro-project.eu/, checked on 21/07/2013.

Smith AJ (2006) Global Manufacturing Trends and Challenges Driving IT Investments in Supply Network Operations: 2006 and Beyond.

SMLC (2011) Implementing 21st Century Smart Manufacturing. Workshop Summary Report. URI: https://smart-process-manufacturing.ucla.edu/about/news/Smart%20Manufacturing%206_24_11.pdf, checked on 5/01/2013.

SOCRADES (2007a) Service-Oriented Cross-Layer Infrastructure for Smart Embedded Devices. Edited by SOCRADES consortium. URI: ftp://ftp.cordis.europa.eu/pub/ist/docs/dir_c/ems/socrades-v1_en.pdf, checked on 21/07/2013.

SOCRADES (2007b) SOCRADES Demonstrators. URI: http://www.youtube.com/user/fp7socrades/feed, checked on 21/07/2013.

Su J (2007) Component-based Intelligent Control Architecture for Reconfigurable Manufacturing Systems. Virginia State University, Faculty of the Virginia Polytechnic Institute.

Sundermeyer K & Bussmann S (2001) Introduction of agent technology into a manufacturing company - Experiences from and industrial project. Wirtschaftsinformatik 43(2): 135–142.

Sura W (2009) Specialisations within EU manufacturing. Eurostat: Statistics in focus 2009, 62/2009. URI: http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-SF-09-062/EN/KS-SF-09-062-EN.PDF, checked on 16/02/2013.

Swantec (2013) SORPAS. Resistance Welding Simulation and Optimization Software. URI: http://www.swantec.com/sorpas.php, checked on 15/09/2013.

Tassey G (2010) Rationales and mechanisms for revitalizing US manufacturing R&D strategies. The Journal of Technology Transfer 35 (3): 283–333.

Terzic I, Zoitl A, Rooker M, Strasser T, Vrba P & Mařík V (2009) Usability of Multi-agent Based Control Systems in Industrial Automation. In: Mařík V, Strasser T & Zoitl A (eds) Holonic and Multi-Agent Systems for Manufacturing vol 5696. Berlin Heidelberg, Springer: 25–36.

Tuovinen L, Talus T, Koponen E, Laurinen P & Roning J (2010) A task-to-method transformation system for self-configuration in manufacturing. Proc 8th IEEE International Conference on Industrial Informatics (INDIN), 2010: 245–252.

Valckenaers P & van Brussel H (2005): Holonic manufacturing execution systems. CIRP Annals Manufacturing Technology 54(1): 427–432.

Valckenaers P, Heikkilä T, Baumgaertel H, McFarlane D & Courtois J (1999) Towards a novel manufacturing control principle. Proc Second International Workshop on Intelligent Manufacturing Systems: 871–875.

van Brussel H, Bongaerts L, Wyns J, Valckenaers P & van Ginderachter T (1999) A conceptual framework for holonic manufacturing: Identification of manufacturing holons. Journal of Manufacturing Systems 18(1): 35–52.

van Brussel H, Wyns J, Valckenaers P, Bongaerts L & Peeters P (1998) Reference Architecture for Holonic Manufacturing Systems: PROSA. Computers In Industry 37: 255–274.

van Dyke Parunak H (1998) What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC. In: Klusch M & Weiß G (eds) Cooperative Information Agents II. Learning, Mobility and Electronic Commerce for Information Discovery on the Internet vol 1435. Berlin Heidelberg, Springer: 1–18.

van Dyke Parunak H, Baker AD & Clark SJ (2001) The AARIA agent architecture: From manufacturing requirements to agent-based system design. Integrated Computer-Aided Engineering 8(1): 45–58.

Veeramani D, Bhargava B & Barash MM (1993) Information system architecture for heterarchial control of large FMSs. Computer Integrated Manufacturing Systems 6(2): 76–92.

Vyatkin V (2011) IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review. IEEE Transactions on Industrial Informatics 7(4): 768–781.

W3C (2006) Extensible Markup Language (XML) 1.1 (Second Edition). URI: http://www.w3.org/TR/2006/REC-xml11-20060816/, updated on 29/09/2006, checked on 21/09/2013.

Warnecke HJ (1996) Die fraktale Fabrik. Revolution der Unternehmenskultur. 2nd ed. Reinbek bei Hamburg, Rowohlt.

Westkämper E (2011) Manufacturing 2030 in Europe. Discussions in Manufuture ISG, URI: http://www.manufuture2011.eu/presentations/Ps1-Implementation%20of%20ManuFuture%20Strategic%20Research%20Agenda/Engelbert%20Westk%C3%A4mper/Engelbert%20Westkamper.pdf, checked on 3/10/2013.

Wohlers T (2009) Direct Digital Manufacturing. Manufacturing Engineering 142(1): 73–81.

Wohlers T (2013) Wohlers Report 2013. Additive manufacturing and 3D printing state of the industry, annual worldwide progress report. Fort Collins Colo, Wohlers Associates.

Wollert JF (2006) Teil E: Bussysteme. In: Gevatter HJ, Grünhaupt U (eds) Handbuch der Mess- und Automatisierungstechnik in der Produktion. Berlin Heidelberg (VDI-Buch), Springer: 475–484.

World Bank (2013) Manufacturing, value added. URI: http://data.worldbank.org/indicator/NV.IND.MANF.ZS/countries/1W-US-Z4?display=graph, checked on 5/01/2013.

Wyns J, van Brussel H, Valckenaers P & Bongaerts L (1996) Workstation Architecture in holonic manufacturing systems. Cirp journal on Manufacturing Systems 26(4): 220–231.

Wysk RA (2009) An Overview of RapidCIM Concepts. URI: http://www.engr.psu.edu/cim/fame/CIMLAB/ie5517_files/frame.htm, checked on 28/08/2012.

XPRESS (2011) Flexible production experts for reconfigurable assembly technology. URI: http://www.xpress-project.eu/, checked on 7/09/2012.

465. Yadav, Animesh (2013) Space-time constellation and precoder design under channel estimation errors
466. Schaberreiter, Thomas (2013) A Bayesian network based on-line risk prediction framework for interdependent critical infrastructures
467. Halonen, Niina (2013) Synthesis and applications of macroscopic well-aligned multi-walled carbon nanotube films
468. Remes, Jukka (2013) Method evaluations in spatial exploratory analyses of resting-state functional magnetic resonance imaging data
469. Oravisjärvi, Kati (2013) Industry and traffic related particles and their role in human health
470. Czajkowski, Jakub (2013) Optical coherence tomography as a characterization method in printed electronics
471. Haapalainen, Mikko (2013) Dielectrophoretic mobility of a spherical particle in 2D hyperbolic quadrupole electrode geometry
472. Bene, József Gergely (2013) Pump schedule optimisation techniques for water distribution systems
473. Seelam, Prem Kumar (2013) Hydrogen production by steam reforming of bio-alcohols : the use of conventional and membrane-assisted catalytic reactors
474. Komulainen, Petri (2013) Coordinated multi-antenna techniques for cellular networks : Pilot signaling and decentralized optimization in TDD mode
475. Piltonen, Petteri (2013) Prevention of fouling on paper machine surfaces
476. Juuso, Esko (2013) Integration of intelligent systems in development of smart adaptive systems : linguistic equation approach
477. Lu, Xiaojia (2013) Resource allocation in uplink coordinated multicell MIMO-OFDM systems with 3D channel models
478. Jung, Sang-Joong (2013) Personal machine-to-machine (M2M) healthcare system with mobile device in global networks
479. Haho, Päivi (2014) Learning enablers, learning outcomes, learning paths, and their relationships in organizational learning and change
480. Ukkonen, Kaisa (2014) Improvement of recombinant protein production in shaken cultures : focus on aeration and enzyme-controlled glucose feeding

Book orders:
Granum: Virtual book store
http://granum.uta.fi/granum/

UNIVERSITY OF OULU P.O. Box 8000 FI-90014 UNIVERSITY OF OULU FINLAND

ACTA UNIVERSITATIS OULUENSIS

ISBN 978-952-62-0365-2 (Paperback)
ISBN 978-952-62-0366-9 (PDF)
ISSN 0355-3213 (Print)
ISSN 1796-2226 (Online)

UNIVERSITY of OULU
OULUN YLIOPISTO