Otto-von-Guericke-Universität Magdeburg

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

MB

FAKULTÄT FÜR
MASCHINENBAU

Agents and SCT based self* control architecture for production systems

Dissertation zur Erlangung des akademischen Grades

Doktoringenieurin

(Dr.-Ing.)

von Dipl.-Ing. Daria Ryashentseva
geb. am 07.12.1987 in Taganrog
genehmigt durch die Fakultät für Maschinenbau
der Otto-von-Guericke-Universität Magdeburg
Gutachter: apl. Prof. Dr.-Ing. habil. Arndt Lüder
o. Univ.-Prof. Dr.-Ing. Hermann Kühnle
Promotionskolloquium am 26.09.2016

Preface

This thesis is the result of some years of the research and has been created during my study and particular workat the Institute of Ergonomics, ManufacturingSystems and Automation (IAF) of the Otto-von-Guericke University Magdeburg.

That was for me a long and “painful” way, and without help of many people finishing ofthis thesis would not be possible. So, I would like to specially recognize some of them here.

First of all, I would like to thank my supervisor Professor Arndt Lüder,who has made this work possible at all,for positive and productive support during these years.

And also I would like to thank my second supervisor ProfessorHermann Kühnle for the second supervision and important corrections.

In addition, I would like thank all employees, colleagues and also the students of the chair of Factory Operations and Manufacturing Systems for their friendship, support and inspiring discussions, especially Nicole Schmidt for the strict remarks that helped to sharpen this work.

Besides, I would like to thank Professor Valery Finaev from my home university for cheering me up during these years.

Also, I am deeply grateful to my parents and sister, and to all my friends that were around me all this time and to my boyfriend. Those people have always been at my side whatever happened and gave me moral support.

Magdeburg, 24.08.2016

Ryashentseva Daria

Declaration of Honor

I hereby declare that I produced this thesis without prohibited external assistance and that none other than the listed references and tools have been used. I did not make use of any commercial consultant concerning graduation. A third party did not receive any nonmonetary perquisites neither directly nor indirectly for activities which are connected with the contents of the presented thesis.

All sources of information are clearly marked, including my own publications.

In particular I have not consciously:

- Fabricated data or rejected undesired results

- Misused statistical methods with the aim of drawing other conclusions than those warranted by the available data

- Plagiarized data or publications

- Presented the results of other researchers in a distorted way

I do know that violations of copyright may lead to injunction and damage claims of the author and also to prosecution by the law enforcement authorities. I hereby agree that the thesis may need to be reviewed with an electronic data processing for plagiarism.

This work has not yet been submitted as a doctoral thesis in the same or a similar form in Germany or in any other country. It has not yet been published as a whole.

Magdeburg, 24.08.2016

Daria Ryashentseva

Abstract

Markets and production are growing at a high speed, trying to please the requirements of consumers.Therefore production control systems need to keep pace with the production complexity that appeared for those reasons. At the same time, concerned customersimposerequirements of the flexibility of production under disturbances and versatility of products. Globalized markets demand new and individualized products. In these conditions, industry and enterprises search an answer to these challenges.

The complete change of the production process organization in the industry usually can´tbe applied, due to time and costs reasons. As an alternative, new solutionsappear,which are built on legacy systems to provide a migration path from old to new structures and are designed to increase the flexibility of existing control methods. To make solutions be self-adaptable to unexpected disturbances, such as environmental disturbances, product structure changes or business competition, the intelligent autonomous industry control systems, similar to the human brain, should be used.

Among the control technologies, the systems wrapped into agents to provide system readiness for all possible occasions and failures in the system stand out. Another method to provide self-optimization of continuous production processes is supervision.

Hence, thisthesisintends to develop a universal,flexible, intelligent approach based on multi-agent architecture. The multi-agent system technology is used to bring control functions into convenient participation that allows receiving products, customized for desires of the end consumers. Due to the agents´ features, the proposed control system has an intellect and due to the supervisory control, it is self-adaptable and self-optimized.

The proposed control architecture consists of five logically and physically separated agents performing different tasks cooperatively. The High-availability Agent is responsible for safety, the Rescheduler Agent is responsible for data processing within the process control, the Executive Agent is applied to interact with legacy control systems, the Supervisor Agent is applied to perform supervisor-tasks, and the Dispatcher Agent deals with the knowledge base to ensure sustainable control.

The control architecture contributes to reach a high product customization and also high quality by a low development and implementation price. Due to the universal features, the proposed control system can be implemented and adjusted on different application cases in the industry.

The thesiscovers the development of the control system and the implementation on ademonstrationsystem. The results of the implementation of the control model are shown in the thesis.

Kurzfassung

Märkte und Produktion wachsen mit hoher Geschwindigkeit und versuchen, denAnforderungen der Verbraucher gerecht zu werden. Daher müssen Produktionssteuerungssysteme mit der entstehenden Komplexität der Produktion mitzuhalten. Gleichzeitig stellen betroffene Kunden Anforderungen bezüglich der Flexibilität der Produktion infolge von Störungen und vielseitigen Produkten. Globalisierte Märkte erfordern neue und individualisierte Produkte. Unter diesen Bedingungen suchen Industrie und Unternehmen eine Antwort auf die neuen Herausforderungen.

Eine komplette Änderung der Organisation des Produktionsprozesses ist jedoch in der Industrie in der Regel auf Grund von Zeit- und Kostengründen nicht möglich. Als eine Alternative erscheinen neuen Lösungen,die auf Legacy-Systemen aufgebaut wurden,um einen Migrationspfad von alten auf neue Strukturen zu schaffen, als Sinnvoll. Diese wurden entworfen, um die Flexibilität der bestehenden Steuerungsmethoden zu erhöhen. Um diese Lösungen selbstanpassungsfähig gegenüber unerwarteten Störungen zu gestalten, wie zum Beispiel, Umweltstörungen, Produktstrukturänderungen oder Business-Wettbewerb, sollen intelligente autonome Industrie-Steuerungssysteme, ähnlich dem menschlichen Gehirn, verwendet werden.

Unter den Steuerungstechnologien heben sich die Systeme ab, die auf Agenten basieren,um Systembereitschaft für alle möglichen Gelegenheiten und Störungen im Systemzu schaffen. Eine andere Methode, um die Selbstoptimierung von kontinuierlichen Produktionsprozessen herzustellen, ist die Supervisory Control.

Daher beabsichtigt diese Arbeit einen universellen, flexiblen und intelligenten Ansatz zu entwickeln, der auf einer Multi-Agenten-Architektur basiert. Die Multiagentensystem - Technologie wird verwendet, um die Steuerungsfunktionen derartig zu kombinieren, sodass Produkte entsprechend der Vorstellung der Endverbraucher angefertigt werden. Aufgrund der Merkmale der Agenten hat das vorgeschlagene Steuersystem einen Intellekt und infolge der Überwachungssteuerung ist es selbstanpassungsfähig und selbst-optimiert.

Die vorgeschlagene Steuerungsarchitektur besteht aus fünf logisch und physikalisch getrennten Agenten, die kooperativ unterschiedliche Aufgaben durchführen. Der

Hochverfügbarkeits-Agent für die Sicherheit verantwortlich, während der Rescheduler-Agent für die Datenverarbeitung innerhalb der Prozesssteuerung zuständig ist.Die Aufgabe des Exekutive-Agents besteht in der Interaktion mit dem Legacy-Steuerungsystem. Der Supervisor Agent wird eingesetzt,um die SupervisorAufgaben auszuführen und der Dispatcher-Agent beschäftigt sich mit der Wissensdatenbank, umeine nachhaltige Kontrolle zu gewährleisten.

Die Steuerungsarchitektur führt durch einen geringen Entwicklungs- und Implementierungsaufwandzur Erreichung einer hohen Produktanpassung und auch einer hohen Qualität. Aufgrund der universellen Eigenschaften kann das vorgeschlagene Steuersystem auf unterschiedliche Anwendungsfälle in der Industrie umgesetzt und angepasst werden.

Die Arbeit umfasst die Entwicklung des Steuerungssystems und dessen Umsetzung auf einem Demo-System. Die Ergebnisse der Umsetzung des Steuerungsmodells werden in der Dissertation gezeigt.

Contents

List of Figures

List of Tables

Abbreviations

AAS	Asset Administration Shell
ABB	Asea Brown Boveri Ltd.
ACL	Agent Communications Language
ADACOR	ADAptive holonic COntrol aRchitecture
AI	Artificial Intelligent
AMProC	Advanced Manufacturing PROcess Chain
ANSI/ISA	American National Standards Institute / International Society of Automation
BDI	Belief-Desire-Intention agents
CAD	Computer-Aided Design
CIM	Computer-Integrated Manufacturing
CPPS	Cyber Physical Production Systems
CPS	Cyber Physical Systems
DA	Dispatcher Agent
DACS	Design of Agent-based Control Systems
EA	Executive Agent
EE	Extended Enterprise level
ERP	Enterprise Resource Planning
EU	European Union
FFMS	Focused Flexibility Manufacturing System
FIPA	Foundation for Intelligent Physical Agents
FMS	Flexible Manufacturing System
GRACE	InterGration of pRocess and quAlity Control using multi-agEnt technology
GUI	Graphical User Interface
HAA	High-availability agent
HMI	Human-Machine-Interface
HMS	Holonic Manufacturing Systems
I 4.0	“Industrie 4.0” paradigm
IEC	International Electrotechnical Commission
IFAC	International Federation of Automatic Control
IFIP	International Federation for Information Processing
IOT	Internet Of Things

JADE	Java Agent DEvelopment framework
KBS	Knowledge-Based Systems
MAS	Multi-Agent Systems
MES	Manufacturing Execution System
OS	Operating System
PABADIS	Plant Automation BAsed on DIstributed Systems
PERA	Purdue Enterprise Reference Architecture
PID	Proportional–Integral–Derivative controller
PLC	Programmable Logic Controller
PROSA	Product-Resource-Order-Staff-Architecture
RA	Rescheduler Agent
RAMI 4.0	Reference Architecture Model for Industrie 4.0
RFID	Radio-Frequency IDentification
RMS	Reconfigurable Manufacturing System
SA	Supervisor Agent
SCADA	Supervisory Control And Data Acquisition
SCT	Supervisory Control Theory
SGAM	Smart Grid Architecture Model
SG-CG	Smart Grid Coordination Group
SODA	SOcieties and infrastructures in the analysis and Design of Agent-based systems (agent-oriented methodology)
SYSML	SYStems Modeling Language
TCP / IP	Transmission Control Protocol / Internet Protocol
UML	Unified Modeling Language
VDE	“Verband der Elektrotechnik Elektronik Informationstechnik e.V.”(Europe’s technical-scientific associations)
VDI	“Verein Deutscher Ingenieure” (Assotioation of German engineers)
VDMA	“Verband Deutscher Maschinen- und Anlagenbau” (Mechanical Engineering Industry Association)
ZVEI	“Zentralverband Elektrotechnik- und Elektronikindustrie e.V.” (German Electrical and Electronic Manufacturers' Association)

Glossary

Note: In order to keep definitions of terms consistent, the terms were defined according to Oxford Dictionary [OxD16]

Adaptation	The action or process of adapting or being adapted
Architecture	The complex or carefully designed structure of something
Changeable	Liable to unpredictable variation
Disturbance	The interruption of a settled and peaceful condition
Flexibility	The quality of bending easily without breaking
Model	A particular design or version of a product
Process	A series of actions or steps taken in order to achieve a particular end
Product	A thing that is the result of an action or process
Reconfiguration	The arrangement of parts or elements in a different form, figure, or combination
Robustness	The ability to withstand or overcome adverse conditions
Schedule	A plan for carrying out a process or procedure
Supervisor	A person who supervises a person or an activity
To control	Regulate (a mechanical or scientific process)

1. Introduction

1.1 Motivation

In the last decades,major changes in the manufacturing environment of developing countries appear.As a result of globalization, the environment becomes more connected and interdependent. Hereupon, markets and production are growing with high speed and control systems need to keep pace with production complexity. Simultaneously, affected consumers impose requirements of products versatilityandmore individualized products,consequently, demanding production flexibility[ElM09]. For example, car production could offer personal comfortable interior of the vehicle or the use of different colors for exterior parts. The variety of the new goods and services will appear due to the connection of people, things and machines [Koc16].To make the production connected means to provide ease in the access to valid information about the system, to collect it for later usage at the appropriate moment, as advised in the new paradigm Industrie 4.0 [IWP14].This is one of the today's challenges. The collected available data can be used to determine how, when and what should be controlled, added or planned during production cycles.

Information gathering from vertical and horizontal value chains can help to make a customer’s order easily understandable and effortless to perform by a production system. Also, it will improve production organization through a facility, logistics, resources and avoidance of human participation in the future. According to the research of the global team of practice-oriented strategy experts called “Strategy&”[Koc16] the digitalization and interconnection of products and services in Germany will bring €30 billion per year revenues and €110 billion annually for the European industry sector. In five years, more than 80% of companies will digitize their value chains. Research says that 25% of the companies surveyed have already achieved a high degree of digitization of their value chains.

Also, the closer work of humans and machines will help to avoid mistakes and in future will result in the predominance of “smart” technologies. For example, currently, in an Airbus airplane factory, operators must follow tens of thousands of steps, with resulting high costs in the case of a mistake. The current process involves more than a thousand different tightening tools. However, Airbus is starting to develop of network-enabled handheld tightening tools, which quickly direct the employees to the appropriate tool which automatically “knows” the next step and sets the right calibration for the particular part that the employee wants to tighten. The smart tools can also provide the recording of the operations to ensure quality control and eliminate manual logging [Dav15].

Also, new tendencies are highlighted in modern business paradigms "maximum gain from a minimum capital to maximum added value from a minimum of resources" due to European Communities in [EuC09] and increase the demand for intelligent, resource efficient factory operation.

Summing up, to cope with the listed challenges, it is necessary to build a reliable, flexible control system which can perform information collection and is able to provide data to reduce the amount of human intervention. At the same time to cope with the production complexity and provide production versatility. Moreover, the methods to rebuild legacy systems with small expenses should be applied. In this context, discern a steady need for new methods, instruments, and tactics to cope commercially with production development processes and ever more complex products [Meh14], to transform from old control systems, or make them more intelligent.

Thus, the aim of the research is to analyze, how to move from legacy systems to a digital "smart" factory with intelligent filling, whichwillproducehighcustomized products for consumer´s desires, which can flexibly deal with disruptions and failures and facilitate optimized decision-making possibly avoiding of human participation according to Industry 4.0 implementation recommendations [ACA13]. Also, the flexible control system has to be developed corresponding the recommendations and demonstrate its operability.

One of standout approaches among the control technologies is the systems wrapped into agents [Lei09]. Multi-agent systems allow receiving products, customized for desires of the end consumers, also providing a system readiness for all possible occasions and failures in the system, because agents can ensure necessary flexibility [VDI2653-1, LeV14]. In this thesis, a universal approach, based on multi-agent architecture, with functions distributed between agents will be discussed.Thanks to the agents´ properties, this will be described later in Chapter 3. Furthermore, this approach will allow reaching the adaptation to different internal and external disturbances in a production system thatincreases overall reliability.

The term “universal” is used for the description of the control architecture which features are common for the different application types in manufacturing. The architecture combines the basic essential functions of the control system for each considered application and can be customized for other applications due to its flexibility.

As nowadays the trend to find common standards in the industry is shown up [RAM15], enterprises will move to a unified architecture, and it may be assumed to be beneficial to use the control system as part of unification trend.

The following paragraph 1.2 will provide research questions as a basis for further considerations within this thesis.The flexibility of presented architecture makes it easy to adopt different parts of the control model, to make the each control process more precise.

1.2 Research Questions

Due to discussed prerequisites, a flexible universal approach for the control architecture must be found. For that reason, now the research questions will be defined.Firstly, this approach should differ from existing analogs, perfecting their features according to research objectives. Secondly, the proposed controller structure should provide following features.

It should:

- have "intelligentfilling" to minimize human participation,
- provide flexibility of control settings and updates to affecting factors,
- possess universalfeatures and functions, aggregated from different application types and which will make it unified,
- be Industry 4.0 compliable,
- be provided by a low-cost change in construction.

Thirdly, the proposed solution should explicate knowledge about the dependencies between products and production systems and within both of them in order to overcome previously described hurdles.

In the end the running application should be developed and demonstrated.

According to it, followingquestions arise.

Question 1. How to move from legacy systems to the intelligent and flexible control system?

This is a general question which includes, from one side, the idea of the development of the control system based on legacy system (Question 3) and, from another side, the migration to the intelligence. So, answering on the first part of this question, the possible way to preserveenterpriseagainst high reconstruction costsand to make a production system flexible through a flexible control, the intelligent add-on on the existing PLC and CNC, which are mostly used nowadays [Fra13], should be created. To answer this question, administrative shell as an add-on,the knowledge-based control approachesand their integration will be discussed.

The second part of the question is about the presence of intelligence and flexibility.Theintelligentcontrol systems imply intellect distributed into parts in the system, where the parts can communicate and perform tasksindependently,usingself- optimization and adaptation, what is based on knowledge-based systems. Also, knowledge and flexibility feature are mutually linked.Knowledgeapplication is the key to unlock the flexible automation`s potential, and at the same time, flexibility in the programmed automation systems will increase intellect of the system[Adl88]. So, the basement of the “smart” is considered in respect to system´s flexibilityfeature. To understand the concept “flexibility”,the following question is raised.

Question 2. How to provide production flexibility in the manufacturing environment?

Manufacturing should be ready to respond to changing production complexity, new customer´s desires and standing disturbances. To answer this question, the “flexibility” term must be defined, and existing surveys on this topic should be regarded. According to the surveys, the possible reachable flexibility grade of the production system for application types should be named. The time of the application of flexibility feature should be defined.

Question3. How to develop the control system that provides flexible production?

By answering this question the control system development steps will be defined and executed. Enterprise organization will be analyzed; also, how the control system is integratedintothe organization layers will be defined.Thecontrol system developmentrequirements to provideflexibilityimplementation will be discussed, namely software and functional requirements.

Question 4. Which functions should possess the control system to be called universal?

This question focuses on the universal structure of the control system for the different application types. To define the common control functions different application types, basic functions and the functions responsible for reaching flexibility will be aggregated and overlapping will be found.

In this work, all of these questions must be answered to get the possibly best architecture by applying flexible control system in different implementation situations. To define the universality of the architecture, the three types of applications from different production types and different production domains were chosen: transportation on a conveyor, heat production and detail production by an industrial robot. They represent three various kinds of production in the sense of machine usage or assistance in production and will be named as “production job”.

The first type is a conveyor, presenting a part of production management in logistics, to assist and maintain the machine. The second type of production service presents chemical production assistance, namely heat or steam production. Moreover, the last “job” is an industrial robot – a general manufacturing robot arm, presenting operational service by machines in production, such as welding, assembling.

The applications are taken from different production domains and differ from each other in some characteristics. This fact will be used to show that a generaluniversal architecture could be created for these different domains and to confirm that it could be possibly used in all other areas. But before that, three application cases will be described more detailed, naming the structure of the processwith the I-P-O-model(germ.EVA-Prinzip) [Cur06], as it is used in the system analysis domain. The model is shown on Fig.1.1. The inputs, outputs of the control system and process itself,including also the disturbances occurring during the production, will be considered in Table 1.1.

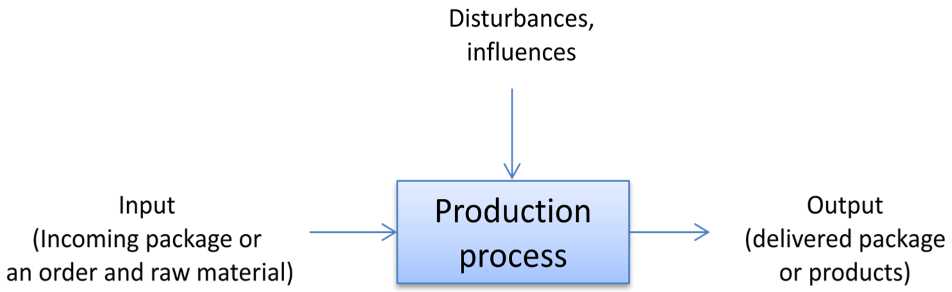


Figure 1.1 The input-process-output model with additional disturbances

The input data for the control system in all 3 cases are the “mode” tables which describe the states of the functioning of each application case, including the parameter settings or programs of the controller, and the “resource” tables in the cases of conveyor and robot. The output data for the control system are eventually changed during the production mode tables and resource tables, reliability data about the process and other process information for the operator, such as transportation parameters. The three application cases are defined in the Table 1.1 to which type of production type they can relate.

In the Table 1.1 also additional rows for thecontrolsystemintegrationdescription are given, namely: investments for the development, theprovidedproduct customization and expected control functions.After the new control system integration, the reachedcustomization in these tree processes are on different levels due to the obtained product: low level in the heater due to customized steam production, middle level of customization of the routing of the product on the conveyor and the high level ofcustomization of the assembled detail on the robot. Investment into the new control system is described with “labor intensive” and “capital intensive”. Labor intensive means a large amount of labor to produce goods or services. Capital intensive means alarge amount of money to produce a goods or services[Cap16, Lab16]. With the investment types, the three production processes can be described as *not labor intensive*and*not capital intensive*, because to integrate the control system it requires a low amount of capital spent on the already existing equipment and a small amount of engaged workersand characterized with low labor contribution of the control system. The expected functions of each production system are to be: dispatching, examination, inspection in all three cases to provide the reliable control and routing/scheduling for the flexible control [Bon98, Lun12].

Later, the relation of architecture to Industrie 4.0, type of provided flexibility, optimization, and levels of automation pyramid affected with integration, possible reconfiguration will be considered.

Table 1.1 Types of application

Conveyor	Heater	Industrial robot
	Process description	
Different objects are being moved on the transporting band of the conveyor. According to the type of the object the destination of it during the identification phase can be defined. **The purpose of the work**isensuring flexibility in path selection taking into account the occupation of the destinations.At the same time availability of conveyor parts is ensured, also providing a path or the resource existence	In the boiler the steam formation occurs. The control is executed by PID controller. Clogging of the boiler occurs periodically, but with frequent changes of supplier clogging fuel occurs more frequently. **The purpose of the work**is ensuring of flexible control depending on the fuel type in the range of states of the heater, also depending on the sensor-data, execution of the suitable control method for the fuel type, also taking into account heater cleaning time, to carry out possible uninterrupted control	Industrial robot assembles the details ina robotic cell of the shop floor. Customers can change the order. **The purpose of the work** is to carry the flexible production process, depending on the desired output product, chooses right set of parts, at the order-incoming time proving its availability, and chooses the right control settings (for example, drilling depth and diameter or point to place glue dots)
	Type of production	
One-off production, batch production, mass-production, continuous flow production	continuous work flow production 24h/d	One-off production, batch production, mass-production, continuous flow production

Input parameters for the production process and for the control		
For the production process: Object on the transportation line **For the control:** Mode tables for the different RFID of objects on the transporting line (possible to give the measurements for the other sensor-type e.g. optic sensors on the line), Recourse tables	**For the production process:** Raw fuel **For the control:** Modetables to describe modes of the heart and assigned fuel type	**For the production process:** Order and material **For the control:** Modetables of detail production, Resource tables
Output parameters for the process and for the control		
For the production process: Delivered object **For the control:** Mode tables, Resource tables, Reliability test, Transportation parameters	**For the production process:** Steam **For the control:** Mode tables, Reliability test, Heater control parameters	**For the production process:** made as desired product **For the control:** Mode tables, Resource tables, Reliability test, Process parameters (coordinates, metrics)
Disturbances		
Lack or occupation of the conveyor equipment, Sensor failure, etc.	Fuel type or manufacturer change, Equipment change, Clogging, etc.	Lack or occupation of the robot equipment, material supply delay, etc.
Provided product customization		
Middle level	Low level	High level
Investment		
Not labor intensive, Not capital intensive	Not labor intensive, Not capital intensive	Not labor intensive, Not capital intensive
Performed functions		
Routing, dispatching, scheduling, examination, inspection	Dispatching, scheduling (of heater modes),examination, inspection	Scheduling, dispatching, examination, inspection

These different kinds of production have the sameflexibilization problems in the control and the more or less same control architecture, based on tables of the resources and modes that can be stored in a knowledge base, with the later selection of the most optimal mode. Each mode is characterized by a set of parameters or programs and provides certain control method [Abr96]. A future distributed control system for an application case should be located all over the hardware on the field level (will be discussed later), showing system resources availability and are assumed to provide control flexibility and product versatility according to the desired outcome.

1.3 Thesis structure

In order to handle the research questions and deal with the three types of applications named above, this thesis is divided into three sections (Fig.1.2). The first section, “state of the art”, shows concepts and definitions required for the understanding. It deals with currently available methodologies, standards and tools, accessible in the area of production and production organization. It shows all modern approaches, how the production system is engineered during the life-cycle of the production process. It consists of the Chapter1, 2 and 3.

The second main section “solution concept” will address the basic approach to deal with architecture. Here, existingcontrol architectures will be presented, integrating different aspects of current technologies, and regularities of communication between agents inside the system, to make it more flexible from both mathematical side in the sense of optimization and engineering side. Also, it includes the approachesthat are used and the analysis of latest papers, such as distributed manufacturing, supervisory, knowledge-based technologies (Chapter 4). Here, the architecture development is directly carried out.

The last section will show how this solution concept could be implemented in the different domains, according to the named problems, where could it be fitted, namely, from the logistics and production domain. Finally, results are analyzed to validate the proposed solution in Chapters 5 and 6. The structure of this thesis is shown in Fig.1.2.

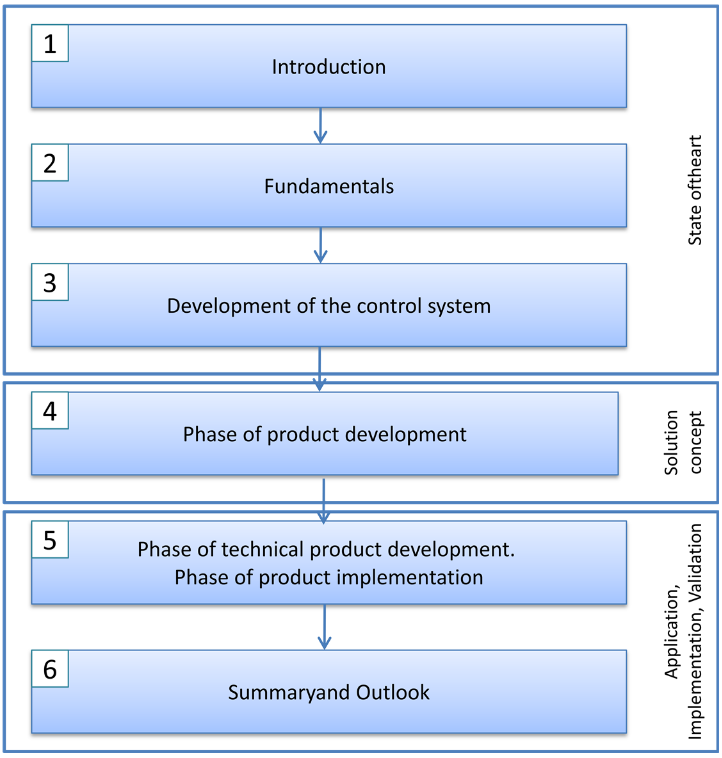


Figure 1.2 Thesis structure

To show more details, here is the description of different chapters with the following core themes.

Chapter 2 will deal with terms manufacturing and production, the changeability and flexibility, going deeper into automation terminology.

The possible standards to develop the control architecture, the automation pyramid transformation, the approaches, methods and models that can be used in the engineering to control a production process, also the valuation of control technologies will be shown In Chapter 3. Also, the system and software requirements, the control system development and system integration must fulfill will be described there.

Chapter 4 will show the DACS methodology [Bus04] applying on three production jobs, resulting in the function distribution between agents. All results will be modeled in Papyrus. The obtained universal architecture will be presented at the end of this Chapter.

Chapter 5 will show the technical side of system development, implementation possibilities and accomplishedattempts.

Chapter 6consists of a summary, analyze of performed work, giving a critical view.

After the main aim of the thesis and questions determination, the next chapters will proceedwiththeanswering on the questions.

2. Fundamentals

The chapter is generally intended to describe all related definitions for the control system. So, all necessary definitions for the intelligent control system development will be presented.Chapter starts with the definitions production and manufacturing. Also, in the chapter the changeability and flexibility features are considered with respect to the defined production services – heater, assembly robot and conveyor. The definition of the Term “flexibility” is described and its range is observed, the particular “focused” flexibility is discussed. Also, new solutions and paradigms, helping with the question how to achieve the required level of the production flexibility are shown.Finally, production control systems will be analyzed.

2.1 Manufacturing and production interrelation

To analyze an intelligent control system development, the life-cycle of manufacturing systems within the production needs to be taken into account. In this paragraph, the terms“Production” and “Manufacturing” and the difference between both terms are defined, because these terms are often confused.

The Dictionary of production engineering [CIR14] claims that despite the conceptual difference between “Manufacturing”and “Production”, in English usage, the term “manufacturing system” addresses not just a single workstation or an individual production department (e.g., foundry, turnery), but also a complete enterprise or a group of enterprises.Accordingly, “Manufacturing” by Segreto and Teti is not only the process making of goods, but also the entirety of relevant technological, economic, and organizational measures directly connected with the processing of materials, i.e., all activities and functionsdirectly contributing to it [CIR14]. But in Europe term “Production” is often used as the “umbrella”term, which includes Manufacturing.

So, according to Engineering textbook[Sko14], Production technology (Fig.2.1) consists of manufacturing engineering (pieces of goods), process engineering (raw materials to products) and power (energy) engineering *(resp. in Germ. Fertigungstechnik, Verfahrenstechnik, Energietechnik)*.

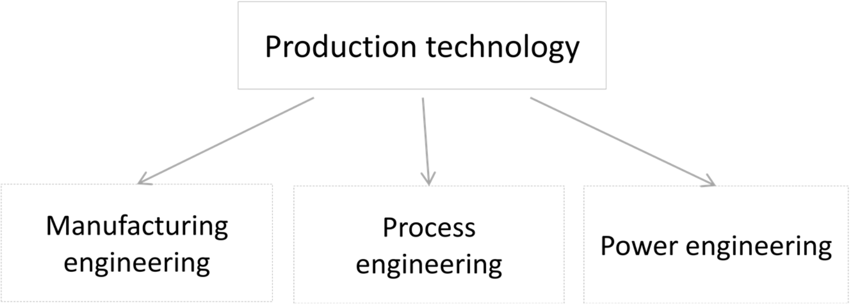


Figure 2.1 Production technology classification according to [Sko14]

Central viewing subject of manufacturing engineering is existing standard of German Institute of Standardization for determiningmanufacturingDIN 8580 [DIN8580] which determines manufacturing processes (in Germ. *Fertigungsverfahren*) as follows: primary forming, reshaping, separating, adding, coating and material properties changing, which considers manufacturing as the specific to processing, not the whole enterprise.

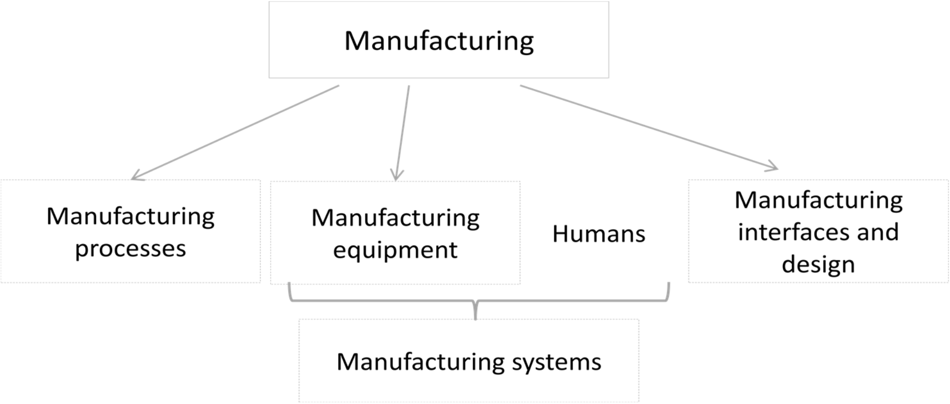


Figure 2.2 Graphical representation of facets of the word “Manufacturing“according to [Chr06]

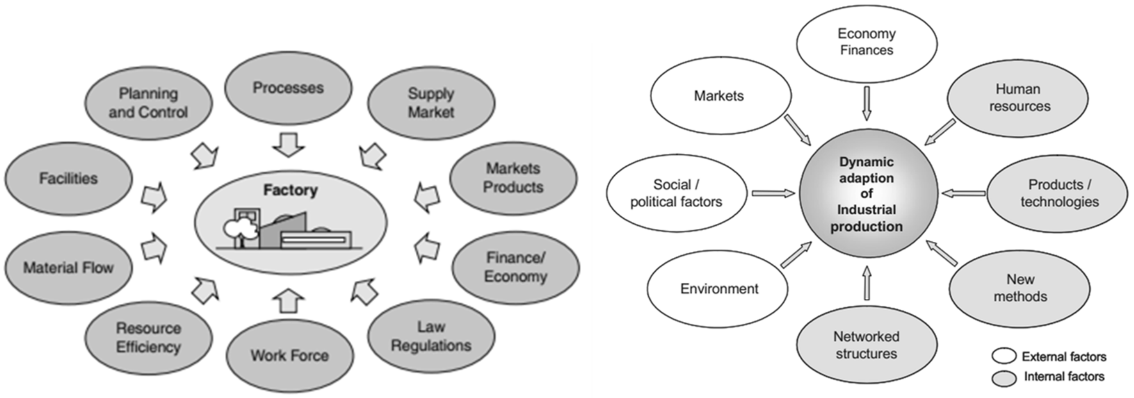
Also, “Manufacturing”by Chryssolouris [Chr06] defined as the transformation of materials and information into products for the human satisfaction needs and has several constituent parts (Fig 2.2):

- *Manufacturing processes,* wherethe form, shape and/or physical properties of a given material are changed,
- *Manufacturing equipment* used to perform processes that in combination with humans represents *Manufacturing systems*, and
- *Manufacturing interfaces* and design, for example, CAD files [Chr06].

Thereafter, due to the fact that three application cases in the thesis correspond to the various types of Production Technology: heater concerns power engineering, conveyor and robot – manufacturing engineering, in the thesis term “Production” will be applied for these cases. Further, analyze of production systems will be carried on.

2.2 Changeable and flexible productionsystems

Production system engineering is experiencing the dynamically changing environment: an appearance of new types of life cycles, a big number of product models and manifold cooperation in networks – all these factors increase the complexity of production processes starting from medium-sized enterprises till highly-complex, as the automobile industry. Stimulated by the influences Changeability and Flexibility of production systems gets in the focus of producing companies [Har12]. So, various influences on a manufacturing system were already analyzed e.g. by Koren, Westkämper, Chryssolouris, Wiendahl and ElMaraghy in [CIR14, Wie07, Kor06, Chr14].



a) b)

Figure 2.3 Influences a) by Wiendahl following [Chr14] b) by Westkämper following [Wie07]

Examples of the influences distribution in different categories are shown on Fig.2.3. So, Wiendahl shows all factors influencing the factory, and Westkämper differs the external and internal factors.

All changeability influences were collected in the thesis and divided into two groups of micro- and macro- triggers (Fig.2.4) as global and more local features, what is indirectly and directly related to the company. Here all social, economic, human- or product- related drivers, all factors that could cause the company to evolve, are named.

Under the influence of these factors, production is changing, growing becomes more complicated, and old control methods no longer cope with the changes in production. As a logical response to these changes, production as a whole and each of its components (as manufacturing and process engineering) must be adapted efficiently and economically, new control systems architecture or new competitive methods must be offered.

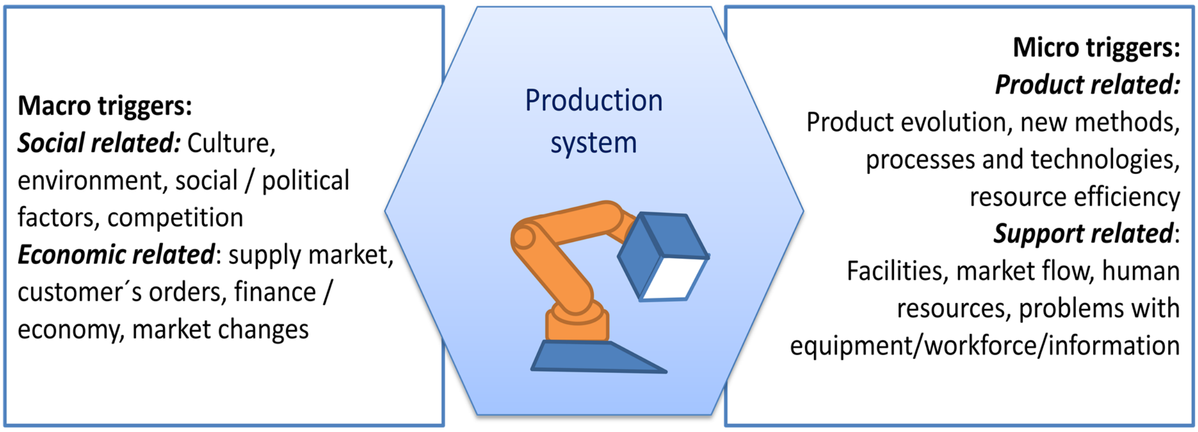


Figure.2.4 Changeability triggers in production systems

Scientists, engineers and even politicians of developing countries [MIC16, Fac13, IND16, IWP14] struggle for a quarter-century with the solution of the production adaptationproblem, as a proofthere are different articles in the book [ElM09], also [Wan07, Wes07]. With the notion, the advancement of production systems, predictable in directions of conversion of equipment and system organizational levels, several paradigms has appeared as the result of it.

For example, some scientists are trying to do reformation from a lower level of production, regarding reconfigurable and flexible manufacturing [ElM09]. Therefore, developed countries at the federal level promote different initiatives – Industrie 4.0, Factory of the Future, Manufuture (see Paragraph 2.3),concerning the entire system. These initiatives have in common the desire to create a manufacturing system, which will show increased efficiency, productivity and better product quality. Herewith with lowest expenditure for equipment, but taking into account the increasing complexity of the system and, as a consequence, operating with the greatest quantity of data. To reach these aims scientists offer a solution: the modern production system should be adaptable to constantly appearing external requirements. So, El Maraghy claims in [ElM09] that for successful participation in dynamic and global production networks production processes, resources, plants structures and production system layouts should be adaptable quickly and with small effort. Scientists have long understood that to adapt quickly, system and/or its structure must demonstrate a property of flexibility, and they started to deal with this issue. Developments in the direction of flexibility started in the 80s [ADl88, Sla83];later, investigation the relationship between the production requirements and the manufacturing flexibility forms continued in 90s [Ton98, Das96]. Search for the best flexible solution lasts till today [Kor06, Ter09, Bah10,Wie07, Wey14], including new initiatives, which are described in Paragraph 2.3.

The term "Flexibility" can be associated with many others terms: agility, versatility, adaptability, reconfigurability. In fact, all these terms are different parts of the core concept of "changeability"(Fig.2.5), which acts as an "umbrella" concept for them.

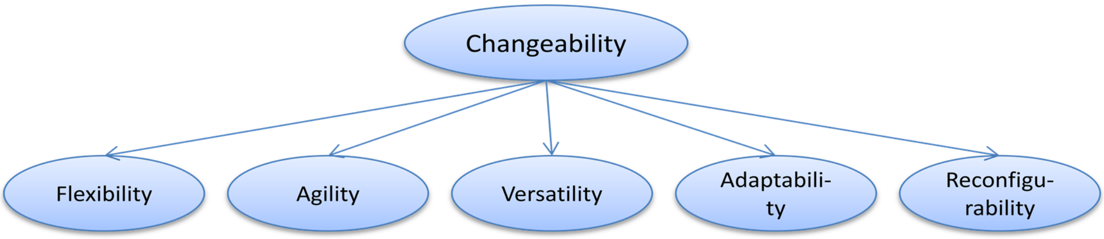


Figure 2.5 Changeability diversity

Not to be confused with definitions, in [ElM09, Das96] the determined meanings of adaptability, changeability; alsoreconfigurable (RMS) and flexible (FMS) system are deliberated and designated.

*"Changeability"*is an ability to withstand the continuous changes and turbulent manufacturing environment facing them, saying ElMaraghy and Wiendahl in [ElM09].

In literature, the term "*Flexibility*" is defined as the ability of a system or facility to adjust them to changes in its internal or external environment [*Das96*]. Also, [Ton98] define "*Flexibility*" as the capacity of a system to adjust itself in response to changing requirements without significant expense in terms of time, effort, cost, or performance. And the term "*Flexible manufacturing*" (in [ElM09]) means manufacturing, which "allows changing individual operations, processes, parts routing and production schedules".

Changeable manufacturing can be seen from the different point of view, it can be expressed through the changing amount of employers´ knowledge, amounts of stations, facilities, products, which allows a system to be changeable and flexible. Wiendahl[Wie07] highlights that the product level so as the production level have to be taken into account, so he presents another changeability classification from Manufacturing point of view, containing five product classes emerged with structuring levels to show the separation of subspecies of a term Changeable Manufacturing (Fig.2.6), it is visible that flexibility occupies a middle position, and is more than reconfiguration and changeover ability, but less than transformability and agility.

This figure is also possible to be used as an example to the customization description due to the attitude between production and product. The customized steam production is a feature, obtained due to the fuel change. It is the lowest level of product- production hierarchy. The customization of the transportation concerns a work piece or sub product on the product hierarchy. The customization of the detail production on the robotic cell is a product or higher according to the hierarchy; that is why this customization is more complex.

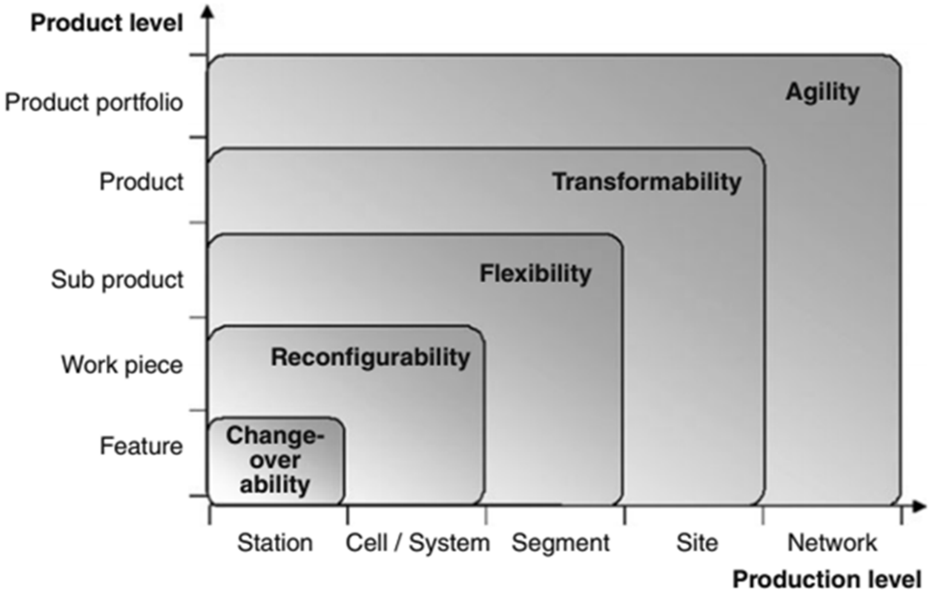


Figure 2.6 Wiendahl´s point of view on Changeable Manufacturing in accordance with [Wie07].

Depending on the changes occurring in the system, “*Flexibility”* feature of the system can be distinguished from “*Reconfigurability”*. So, by Tolio´s et al. view, if the system changes its behavior by changing its configuration, this is not a flexible system, this is a *reconfigurable system*. If the system can change its behavior without changing its configuration, this *system is flexible*[Tol06]. For example, changeover ability is an ability to change over from one part to another in parallel, when the operation is performed on another piece. So, this is a typical example of the flexible manufacturing system (FMS).

Koren describes *Reconfigurable Manufacturing System (RMS)* as a system designed at the outset for rapid change in structure, as well as in hardware and software components, to quickly adjust production capacity and functionality within a part family [Kor06]. In his work he distinguishes dedicated, RMS and FMS in the table by some fixed or adjustable parameters. He shows that FMS have a fixed machine structure; it is possible to agree with him regarding that the system will be still adjustable, allowing flexibility. For example, to avoid reconstruction changes, adding all the time and reconstructs parts of plant, flexibility can be realized in the software changes. Contested, that the productivity of FMS will be low, but his words can be attributed to that fact, that after 10 years after publication of his survey new technologies have appeared, and now the highest productivity of the system can be reached, using new software application on the legacy systems, providing horizontal integration, what Industrie 4.0 requests.

Under the conditions of the construction of new systems in the presence of modern equipment control technologies, the design flexibility of the system can be anticipated beforehand, forcing investing vast amounts of money in the development to foresee the possible level of flexibility of different production systems but just at that time interval. This “pre-paid” way is not justified because of the rapid development of new control methodic, driven by paradigms (Industrie 4.0, etc.) and that will bring the necessity of system reconfiguration anyway. Also, Koren claims that since the machine builders do not know the particular use of the machine when they design it, flexible systems and their machines are constructed with all potentialfunctionality built in. The completefunctionality is often underutilized and constitutes a capital waste [Kor06].

Another possibility “postpaid” allows staying on legacy systems with the later software re-design, if there is an opportunity that always will allow using latest software updates without system reconfigurability or with fewer expenses. This way is justified for small and middle enterprises, because in the case of a big and valuable product, for example, the highest flexibility of a car assembly line will be difficult to reach by updating software, here the resource reconfiguration is needed. But drilling machines flexibility or a heater control flexibility can be classified under this category of flexibility.

The term flexibility has been widely studied, different authors allocate own flexibility types. The identified flexibility types typically refer to various elements and attributes of a production facility such as machine, product, processing, operation, routing, capacity, expansion, failure, design and system.So, Sanchoy K. Das in [Das96] defines such types as machine, routing, process, product, and volume flexibility.

ElMaraghy and Caggiano[CIR14] consider also more flexibility types: production, mix, expansion. The Table 2.1 presents both classifications.The classifications intersect at some points; so, they name with the process flexibility and production flexibility the same thing.

As Adler has marked in [Adl88],*Flexibility* is now a hallmark of automation. In his article, he also brings the survey-classification-table with linking just two key dimensions of flexibility: process and product, and separated between these two types other classifications from Mandelbaum, Gerwin, Buzacolt, Browne and Jakumar.

So, Mandelbaum name the product and process flexibility as state and action flexibility and Gerwin refers volume flexibility to the process flexibility dimension. Adler claims that from engineers’ point of view, the process dimension is the most “exciting” one, because of its “growing ability to design into machine systems sufficient flexibility and intelligence to make them far more robust about to a broad spectrum of process contingencies”. From the societal and managerial points of view, however, the bigger challenges and opportunities seem to derive from the development of flexibility about changes in the product dimensions.

Table 2.1 Flexibility classifications by Das and ElMaraghy

Type	Sanchoy K. Das	ElMaraghy and Caggiano
Machine Flexibility	various operations without requiring a prohibitive effort in switching from one operation to another	various operations performed without set-up change
Routing Flexibility	alternate routes to manufacture a product through the system	number of alternate routes considering all part types/number of part types
Process Flexibility	the set of product types that the system can produce without major setups	-
Production flexibility	-	number of all part types that can be produced without adding major capital equipment
Product Flexibility	is the ease with which new products can be added or substituted for existing products	ease (time and cost) of introducing new products into an existing product mix
Volume Flexibility	is its ability to be operated economically at different overall output levels	the ability to vary production volume profitably within production capacity
Mix flexibility	-	ability to change product mix without affecting the production quantity
Expansion flexibility	-	ease (effort and cost) of augmenting capacity and capability when needed (approaching reconfiguration)

Some scientists propose the customization of system flexibility and claim that it provides economic advantages in terms of system investment costs [Ter09], for example, a class of production systems called Focused Flexibility Manufacturing System (FFMS) [Tol06]. FFMS differ from RMS in the timing of flexibility acquisition and explicit analysis of the cost of flexibility. The idea of RMS is to provide in each moment the production system exactly with the capabilities required by the production problem and modify the system if needed. FFMS consider reconfigurability and flexibility as two options and a mix of them by their costs [Ter09].

It is obvious, that in conditions of the globalized market, which needs the flexibility of the production system, and a variety of the products, the choice of the creation of Flexible control system is justified. So,**the idea of the research is to work with legacyproduction systems, making them flexible through control software update** (with a choice of a right control system, for example succumbing to the stream using smart shells from Industrie 4.0 paradigm), **in the absence of the possibility of changing machine structure**. The chosen application cases now will be classified with a given flexibility types from different authors.

After conducted analysis of flexibility types,it is feasible to claim that achieving ofa flexible production system, which will respond on changing production complexity, new customer´s desires and standing disruptions,is possible providing process or product flexibility, manifested as routing-, volume-, mix flexibility. The classification of pursued types of flexibilityfor the three application cases is presented in Table 2.2.

Table 2.2 Pursued Flexibility for chosen application

Conveyor	Heater	Robot
Legacy system		
Fixed machine structure, software update is required	Fixed machine structure, software update is required, limited to the replacement of sensors and actors	Fixed machine structure, software update is required
Pursued Flexibility type by Adler [Adl88]		
Very product- oriented, less process- oriented	Very process- oriented (due to steam production), less product oriented	Very product- oriented, less process-oriented
Pursued Flexibility types by Das[Das96]		
Process flexibility, Product-, Routing flexibility	Process Flexibility, particularly Product flexibility, Volume flexibility	Machine flexibility, Volume flexibility, Process-,Product Flexibility
Pursued Flexibility types by ElMaraghy[CIR14]		
Volume flexibility, Routing flexibility Mix flexibility, Expansion flexibility	Product flexibility, Expansion flexibility	Machine flexibility; Production flexibility, Mix flexibility, Product flexibility

Thus, the control system based on legacy production system will provide in the cases of manufacturing engineering (robot and conveyor) more product oriented flexibility and in the case of the power engineering (heater) - product oriented flexibility. The flexible production system will be achieved due to Process-, Product-, Routing flexibility in the conveyor, Process Flexibility, particularly Product flexibility and Volume flexibility in the heater and Process-, Product -, Machine- and Volume Flexibility in the assembly robot.

The description of modern paradigms, aimed at creatingflexible production systems described above will be listed now.

2.3 Initiatives in the direction of flexible manufacturing system

As aforementioned, nowadays there is a bigger need of flexible production in many domains than before. In logistics, for example, it is reasonable to adapt the speed of conveyor belt fast and without reconfigurations to the different types of goods, in the power plant – to adapt control to the fuel change, in an assembling line – to adapt the setups of assemble robot to manufacturing new products. Unfortunately, in the production systems the adaptation to changeable conditions is not enough implemented. That is why engineers are struggling with the search for new methods to increase the quality of the production that will shorten the life-cycle of the product, decrease products waste, and get more gain in the production. A solution of the problem would be to make production systems more flexible and to find new methods to support this idea.

The government of different countries pushes the idea to createquickly the flexible production. That’s why in different countries a tendency of creation the new lines in industry and engineering, can be traced emanating from the government; appear new engineering initiatives, as the response to nettling challenges. In the 21st century under the rapid development of technologies,having all the means at hand to make the sustainable future closer, the EU initiative Manufuture2020 [MAN16, Wes07] is to implement sought-for methods in real life. Again under the trend of 4th Industrial revolution, in Germany within the Industrie 4.0 initiative the topics of data consistency, integration and flexibility are considered [ACA13, IWP14, Poe14]. The "Industrial Internet Consortium" in the United States [IND16], the "Factory of the Future" in France and the United Kingdom [Fac13] or the "Made in China 2025" strategy of the Chinese government [MIC16] pursuit similar goals. In Italy, the FabbricadelFuturo project (2011-13) supported research initiatives in areas including customization of products, reconfigurable factories, high performance and sustainability [EuC09].

Here, the German initiative the Industrie 4.0 will be described more precisely, as the German engineering is often a driver of innovation in production technology and equipment in the world [BBC13, ACA13].

2.4 Industrie 4.0

Every technological leap in the industry accepted to name “Revolution”: the so-called 1st industrial revolution was in the field of mechanization with the creation of first steam engine in 1853, 2ndindustrial revolution of the intensive use of electrical energy and auto production, and 3rdindustrial revolution responsible for the widespread digitalization and first programmable logic controllers. "Industrie 4.0" (I4.0) term was established for a "4th industrial revolution" (Fig.2.7) [ACA13].

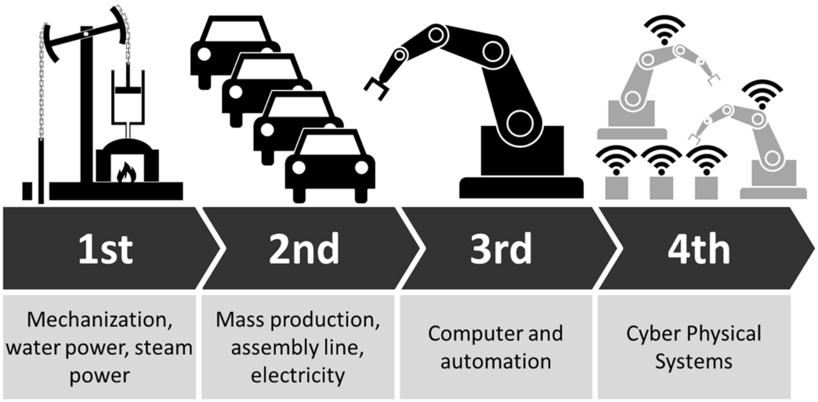


Figure 2.7. Technological revolutions according to [Mar16]

Status Report from VDI/VDE-Gesellschaft Mess- Und Automatisierungstechniktechnical committee [ISR16] describes Industrie 4.0 as the scenario of industrial production, which is characterized by three aspects:

- a new level of organizing and controlling the entire value chain with the life cycle of products , that means comprehensive digitalization of the horizontal and vertical value chains,
- the availability of all relevant real time information which is achieved by interconnecting all instances that participate in the value creation processes,
- the creation of dynamic, real-time optimized and self-organizing cross-company value networks by interconnecting humans, objects and systems, and their abilities.

Horizontal and vertical value chains are presented on Figure 2.8.

According to [Koc16] in five years', 85% of companies will implement Industrie 4.0 solutions in all important business divisions. They expect that 80% of the vertical and 86% of the horizontal value chains will have a high degree of digitization by 2020 and will therefore be closely integrated.

This integration has two sides: the functionalities of industrial automation systems must be available for use by the overriding management systems and value creation management functions must have the capability of being integrated into the systems of industrial automation [Sch15]. Taking into account these demands they created the reference architecture for the industrial automation systems. The reference architecture model Industrie 4.0 (Fig. 2.9) provides thebasic reference architecture for an I4.0 system.

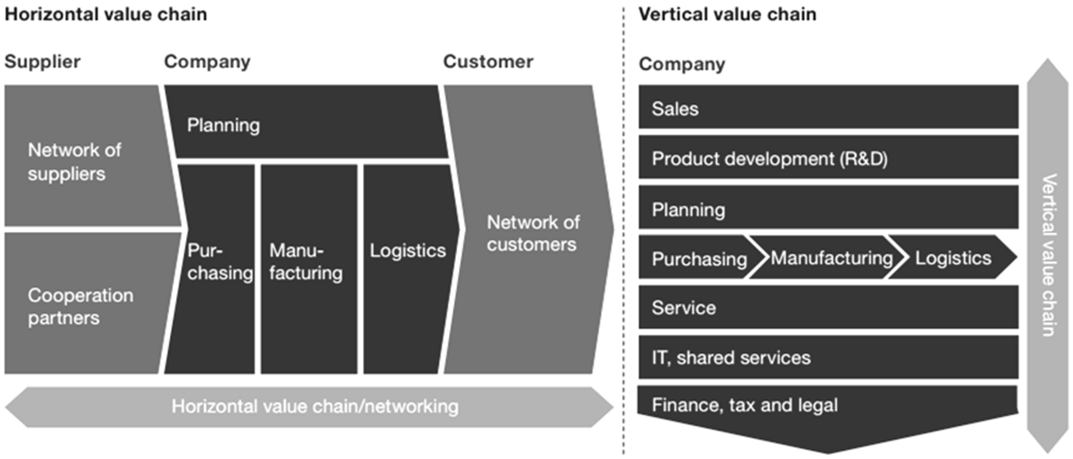
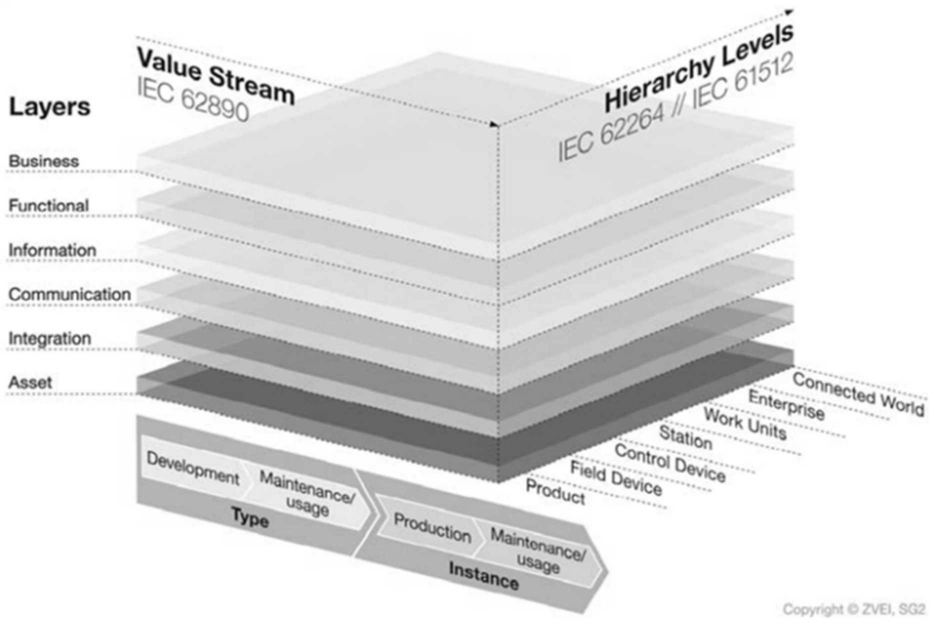


Figure 2.8 Horizontal and vertical value chains according to [Koc16]

The reference architecture model for Industrie 4.0 [RAM15] is the result of the cooperation of different institutions, such as VDI/VDE Society Measurement and Automation Technology (“VDI/VDE-Gesellschaft Mess Und Automatisierungstechnik”), Bitkom, VDMA and ZVEI and is best suited to represent I4.0 space. The vertical axis represents the various perspectivesof the enterprise, such as data maps, functional descriptions, communications, assets and business. The horizontal axis represents value stream. And the third axis represents a functional hierarchy, and not the equipment classes or hierarchical levels of the classical automation pyramid [StR15].



Copyright © ZVEI, SG2

Figure 2.9 RAMI 4.0 according to [ISR16]

I 4.0 describes the term “Technical assets” as an artifact produced especially to fulfill a role within a system. In the document different categories of the technical assets (are shown on Fig.2.10): immaterial assets (meta-models, planning assets, empirical assets) and material assets (physical assets), are described, more information is presentedin [ISR16].

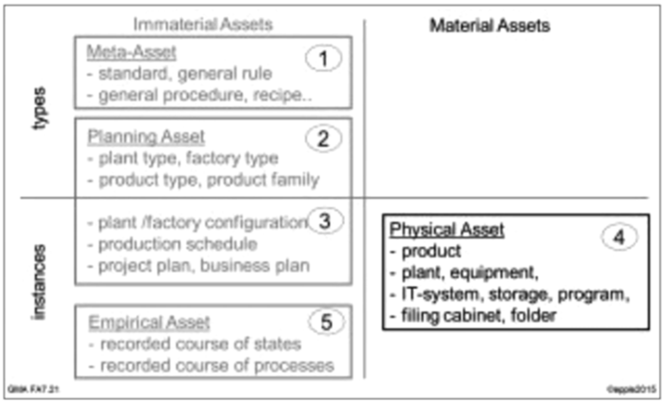


Figure 2.10Categories of technical assets according to [ISR16]

In this way the software product (e.g. the control system) and product obtained during production (e.g. detail) are the physical assets. Assets which have assigned their own objects (identities, states and life cycles) in the information world for their administration also called “Entities” [ISR16] per scale “degree of awareness” according to Status Report [StR14, StR15]. Also, in [StR15] other “degrees of awareness” are distinguished: Individually known item (have a unique system-wide known name), Anonymous item (indirectly identifiable) and Unknown item (unknown for information world).

All assets are characterized by the common life-cycle schema (Fig.2.11) and value course (Fig.2.12).The asset is designed, produced, used, maintained, re-engineered and removed. It doesn’t matter if it is a product like produced steam or a constructedplant, they are technical assets, whichsimilarlife-cycles are presented on (Fig.2.13). On Fig.2.13 after the Usage stage could be added additional stages: re-engineering and removal, according to Fig.2.11.

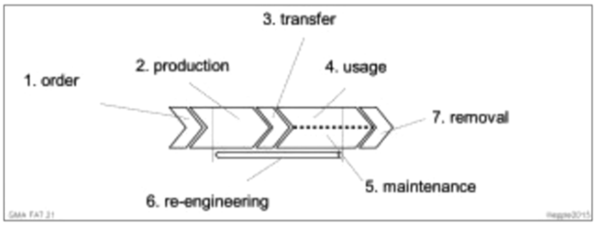


Figure 2.11 Technical asset standard life cycle according to [ISR16]

Applied to the life-cycle schema, technical assets show a similar value course (Figure 2.12) that is obtained due to Horizontal and vertical value chains (Figure 2.8).

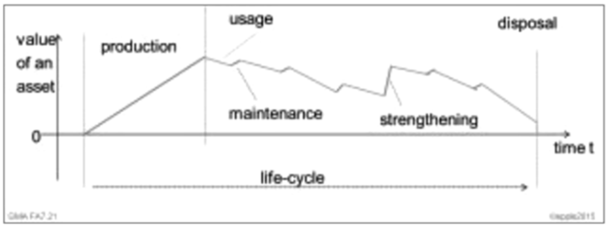


Figure 2.12 Typical value course of a technical asset according to [ISR16]

Looking on the Figure 2.12 and 2.13, it is obvious that flexibilization of the technical assets (software or equipment) can be either planed on the first stage before production or added on the strengthening stage or during the re-engineering. To be more precise, the standard VDI 5200 (*Blatt* 1 “Fabrikplanung”) [VDI5200] defines factory planning as a systematic process consisting of planning phases and implementation phases.

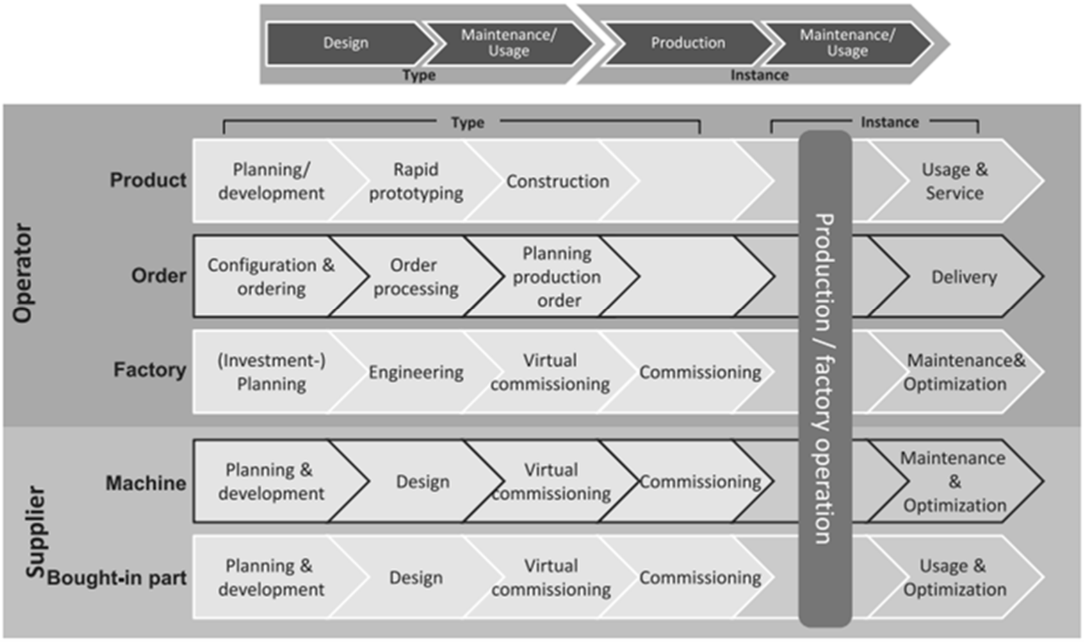


Figure 2.13 Relevant life cycleaccording to [StR15]

Due to [VDI5200] it is divided into goal-oriented phases, conducted with the help of different methods. Looking at a life-cycle of a factory there are plan-cases to mention:

- New planning: planning a factory from the beginning.
- Re-planning: enhancements, optimization or adaptation planning of the existing factory.
- Deconstruction: demontation of the factory.
- Revitalization: reactivation to a new industrial use of the factory.

It is evident that flexibility inside the factory, considering it inside of the new control system, is possible in the cases: new planning, re-planning and revitalization of the factory. On the first stage, the expected flexibility can consider already possible future reconfiguration capabilities. Applicable to the aim of the thesis, if the flexibility of production was not planned from the begging, it can be added either on the strengthening phase, using equipment renewal, or software update over the legacy system, what is cheaper. It spreads on other assets, due to the fact that they have the same life-cycle.

The software update of the production system is described by Industrie 4.0 as a physical asset (Fig.2.10), received during the programming, as well as the equipment. Also, the scheduling represents planning asset, the states of production represent empirical assets. The Asset Administration Shell (AAS) is a concept to organize the administration of such assets within the digital IT system. The AAS and its asset connected by the digital communication system form an Industrie 4.0-component. Representation of Administration shell and “degrees of awareness” are showninFig. 2.14.

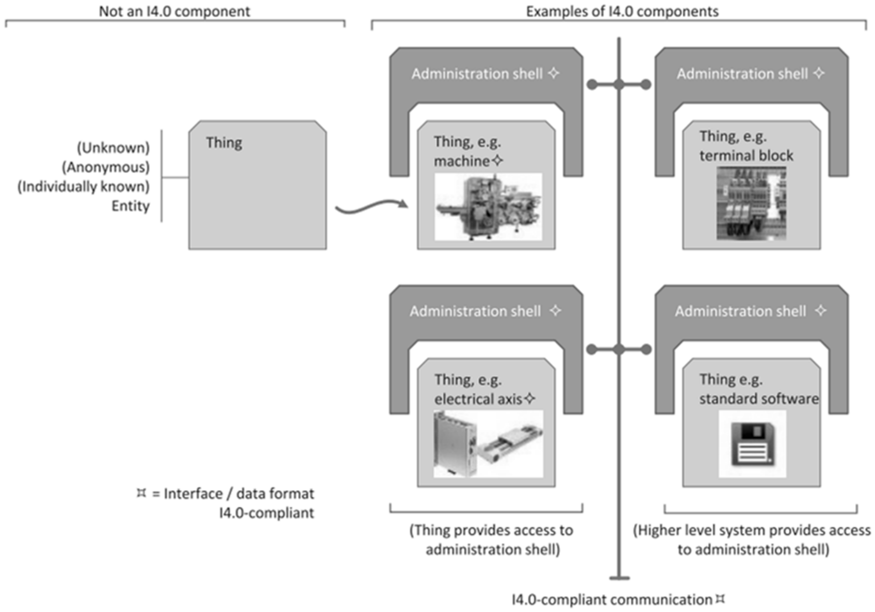


Figure 2.14 Transformation of the object to Industrie 4.0 - component according to [StR15]

Kagermann et al. describes in [ACA13] big potential of Industrie 4.0applied to individualization of customer requirements (Industrie 4.0 allows for accommodating individual customer-specific criteria for design, configuration, ordering, planning, production and operation including short-term change requests), also flexibility to disruptions (dynamic business and engineering processes enable last-minute changes to production),optimized decision making andresource productivity and efficiency.

[ACA13] says that in the future, businesses will establish global networks that incorporate their their warehousing systems, machineryand production facilities in the shape of new systems - Cyber-Physical Systems (CPS). CPS comprise smart machines in the storage systems, manufacturing environment and production facilities capable of autonomously exchanging information, triggering actions and controlling each other independently. By A. Lee CPS determined as “embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa” [Lee08,CPS13].Broy gives in [Bro10, Bro13] following key capabilities of CPS:

1. The collection and acquisition of data such as parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time.

2. CPS covers comprehensive functionality through a systematic selection, incorporation and use of services. Also services in the cloud, service directory, composition heading autonomic behavior and self-organization.

3. Self- organization and adaptation is reached by self-awareness about the systems’ own situation, status and options for action. This requires also modified work processes, logistics processes, habits, interaction, etc.

4. CPS implies distributed, cooperative and interactive perception and evaluation of the situation. That requires subsequent coordinated assessment and negotiation of the decision. To achieve it, decision-making on the basis of uncertain knowledge, cooperative learning and adaption to situations is required.

5. CPS expects intuitive, multimodal, active and passive HMI-support, as evaluation of data is required both in a system and human concept.

CPS is the basis of I4.0 for production automation and to achieve I4.0 the necessary characteristics of CPS [Vog14] are:

- architecture model,
- communication and data continuity in engineering and at runtime,
- intelligent products and production systems as well as and
- the data integration and data preparation for humans.

Relation of three chosen production applications to the requirements of Industrie 4.0 and CPS according to the planed purposes in Chapter 1 isgivenin the Table 2.3.

The control architecture which will be presented in this thesis is created to consider different influences on the process, disruptions and disturbances. Increased production quality and robustnessare possible when all elements in the system can be able to know information about each other, are available and accessible.

Table 2.3Features of the control system conforming to CPS and I40 requirements

Conveyor	Heater	Assembly robot
Industrie 4.0 conform		
• Routing customization and resource awareness due to information availability and accessibility • Increased production quality and robustness, due to flexible responses to disturbances • Reduced production time due to appropriate destination scheduling • Flexibility in path selection taking into account the occupation of the destinations	• Customized steam production • Energy optimization and lower emissions due to smart sensors • Increased production quality and robustness, due to flexible responses to disturbances • Reduced environmental impact	• Assembled detailwith customer specific design and production • Resource awareness due to Information availability and accessibility • Increased production quality and robustness, due to flexible responses to disturbances
CPSconform		
• Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time • Self- organization and adaptationconsidering resource accessibility (part of conveyor are available) and the object´s type • Cooperation with humans in informing and control	• Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time • Self- organization and adaptation considering customers' desires (fuel selection) • Cooperation with humans in informing and control	• Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time • Self- organization and adaptation considering customers' desires and resource accessibility (assembled part will be created by chosen instructions and instruments) • Cooperation with humans in informing and control

The control system should possess the knowledge about system and production process to be flexible. Fulfilling all features listed in Table 2.2 the control system will be flexible and will try to meet the Industrie 4.0 and CPS requirements. The featuresnamed in Table 2.2 to achieve can be realized with production control placed in a field control level and the logic control level, this will be chosen in next chapter. In this way, theresearch question number 2 is answered.

After analysis of prerequisites and modern paradigms for control system creation, the standards, steps and requirements for the control system development in the next chapter will be discussed.

2.5 Positioning of the control system

To understand the position of the intelligent flexible control system (Question 1 and 3), the enterprise structure will be researched. Any enterprise includes management, logistics, and field control. To know how it is organized, different representations of automation pyramids will be considered.

As one view on enterprise, PERA (the Purdue Enterprise Reference Architecture) is a complete Enterprise Reference Architecture, as defined by the IFAC/IFIP Task Force on Enterprise Integration from 1993 [Wil93], and is used to illustrate most of the concepts and tasks involved in enterprise integration. This architecture defines the interfaces are those which permit the physical flow of either material or energy or both across the boundaries between the individual units of the manufacturing equipment (Fig.2.15):Interfaces between the Manufacturing Equipment and Information, Interfaces between the Manufacturing Equipment and Human and Organizational Architectures, Interfaces within the Human and Organizational Architecture, Interfaces between the Information System and Human and Organizational Architectures, Interfaces within the Information System Architecture.

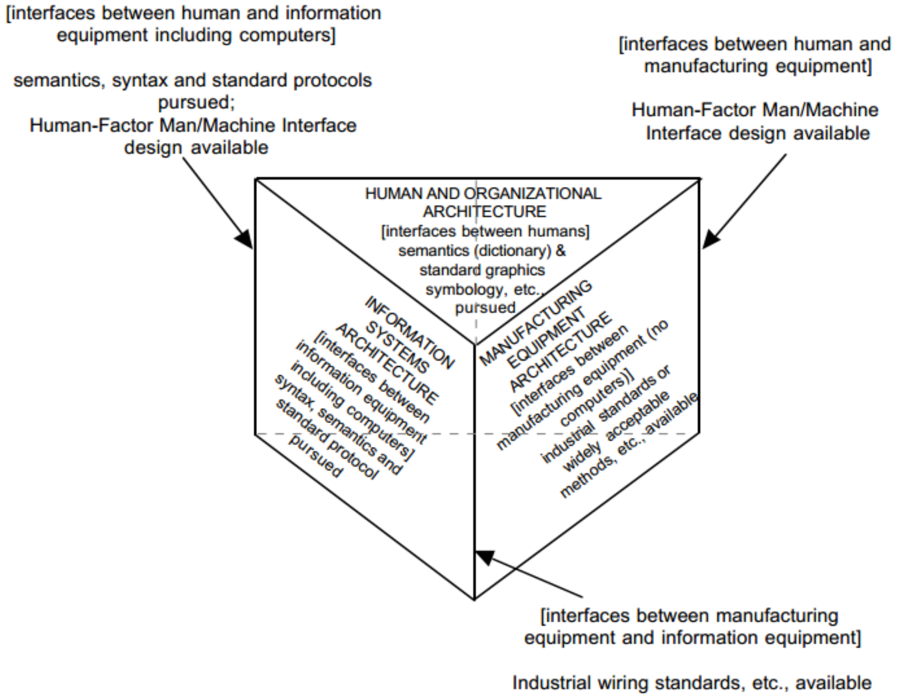
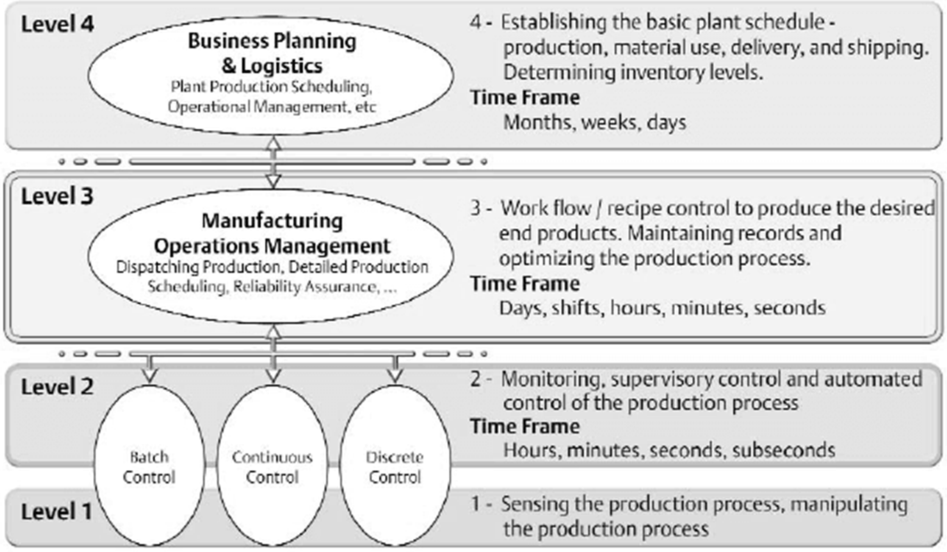


Figure 2.15 Representation of intra-architectural interfaces in PERA according to [Wil93]

According to ANSI/ISA 95 Standard “Enterprise- Control System Integration”, developed by ISA Committee of volunteer experts [ISA95], there are following levels in the enterprise. The bottommost level is the physical production process with batch-, continuous-,or discrete control. Higher, manufacturing operation management is placed. And the highest level is “Business planning and logistics”on Fig.2.16. In this picture, the time frames can be seen, which can be used to describe the information dependencies from seconds till years.

Despite differences in presentation, after all, picturing of the enterprise structure is mostly the same. In the lowest level, always sensors, actuators are placed over the production process.



ISA95: Enterprise-Control System Integration 1: Models and Terminology; ANSI/ISA95.00.01– 2000

Figure 2.16 Enterprise integration architectureaccording to [ISA95]

Then over it always a controller PLC or SCADA system are located, then higher more complex systems, such as MES and ERP, are located. The typical presentation of it, for example, is from Schmalz organization[SCH16], one of the suppliers of the automation technology (Fig.2.17).

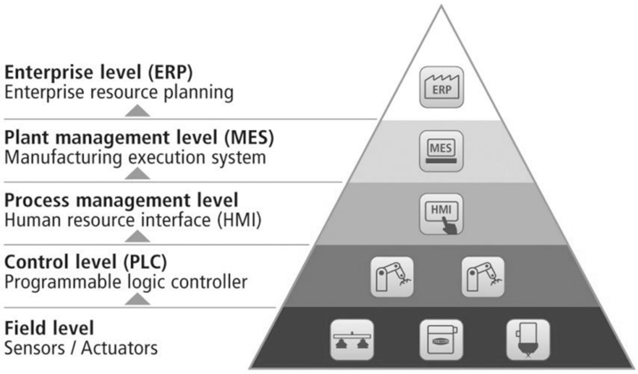


Figure 2.17 Automation pyramid according to Schmalzorganization [SCH16]

In these pyramids there is a typical tasks distribution:

- Rarelyhighlighted Enterprise Network or Extended Enterprise level (EE): High-level reconfiguration, all members of the supplier and distributionchains, which share a common goal of obtaining market shares through the product realization [Kue15].
- ERP systems are always responsible for planning, production, development, manufacturing, sales and marketing, human resources, inventory, etc. They focus on customers.
- MES are responsible for managing and monitoring work-in-process on a factory floor, for execution considering finite scheduling, production dispatching, schedule execution, data collection, and quality management, so, it ensures process transparency and maps information within the supply chain, etc. [VDI5600].
- Process control (from SCADA to process) focus on process operations, sequencing, work instructions and maintenance.

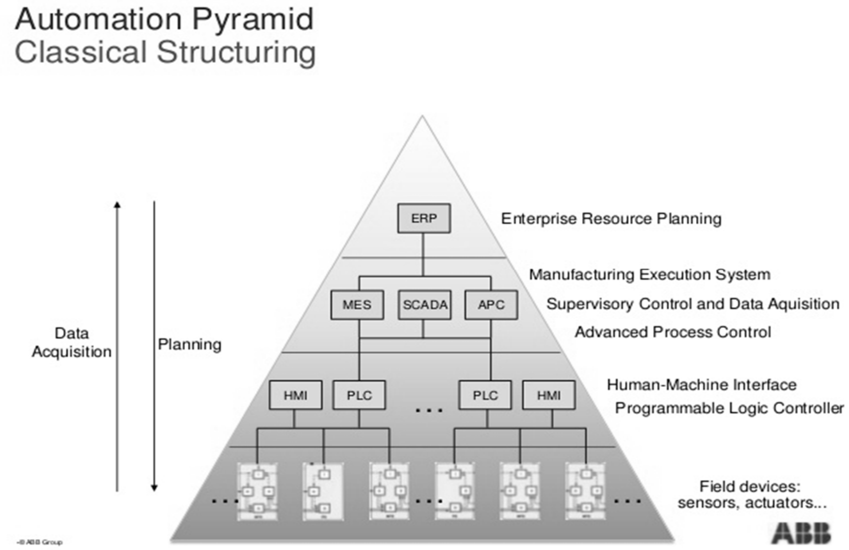


Figure 2.18 Automation pyramid according to [ABB16]

The ABB Company[ABB16]version also concentrates on the process view of an organization, showing in which direction data acquisition occurs. They merged SCADA level and MES level, but the enterprise planning still starts from resource planning, management, control to the sensors on the last level (Fig.2.18).

Vogel-Heuser et al. have suggested another type of pyramid [Vog15] using the ERP and field levels (Fig. 2.19), but between them,all other components presented as Information transformation, from the process level tillMES. As obvious, here the information plays a great role in the system and moves inside one system or more.

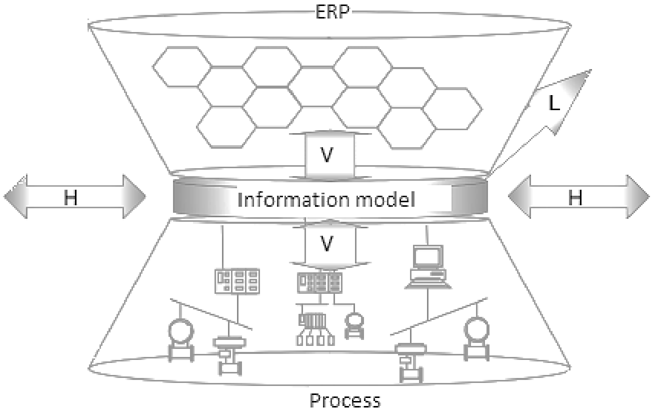


Figure 2.19 Automation pyramid from Vogel-Heuser according to [Vog15]

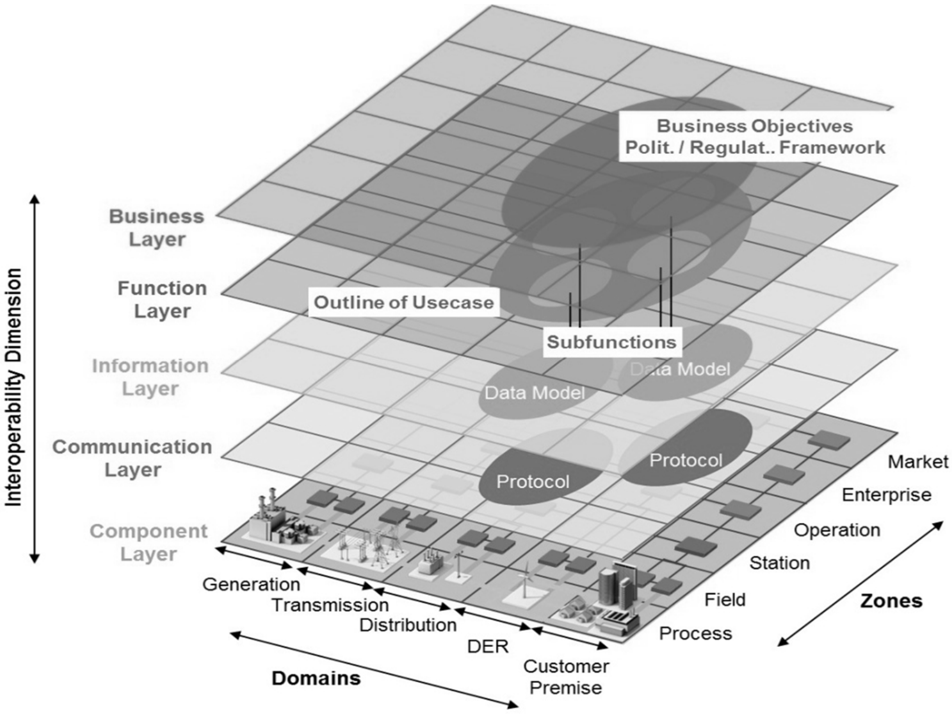


Figure 2.20 Smart Grid ArchitectureModel according to [SGA14]

One more complex pyramid,which is a prototype of the pyramid, presented by the Industrie 4.0 Initiative, the RAMI 4.0, the Smart Grid ArchitectureModel (SGAM) which has been defined by the European Smart Grid Coordination Group(SG-CG), showed in Fig.2.20. [SGA14]. SGAM has been adjusted on the basis of I4.0-requirementsand expandedto the form on Fig.2.9.

Considering all mentioned pyramids and information distribution, after all, the automation pyramid can be presented in the 3-d version (Fig.2.21), uniting information distribution (presented in color saturation), timeand perspectives of an enterprise based on

RAMI 4.0 (horizontal axis). On the vertical axis arethenine hierarchy levels of the modern enterprise according to [Roe16]. Thethird axis is time, which in future will be reduced to a minimum; that is why on the picture on every level it tends to a minimum.

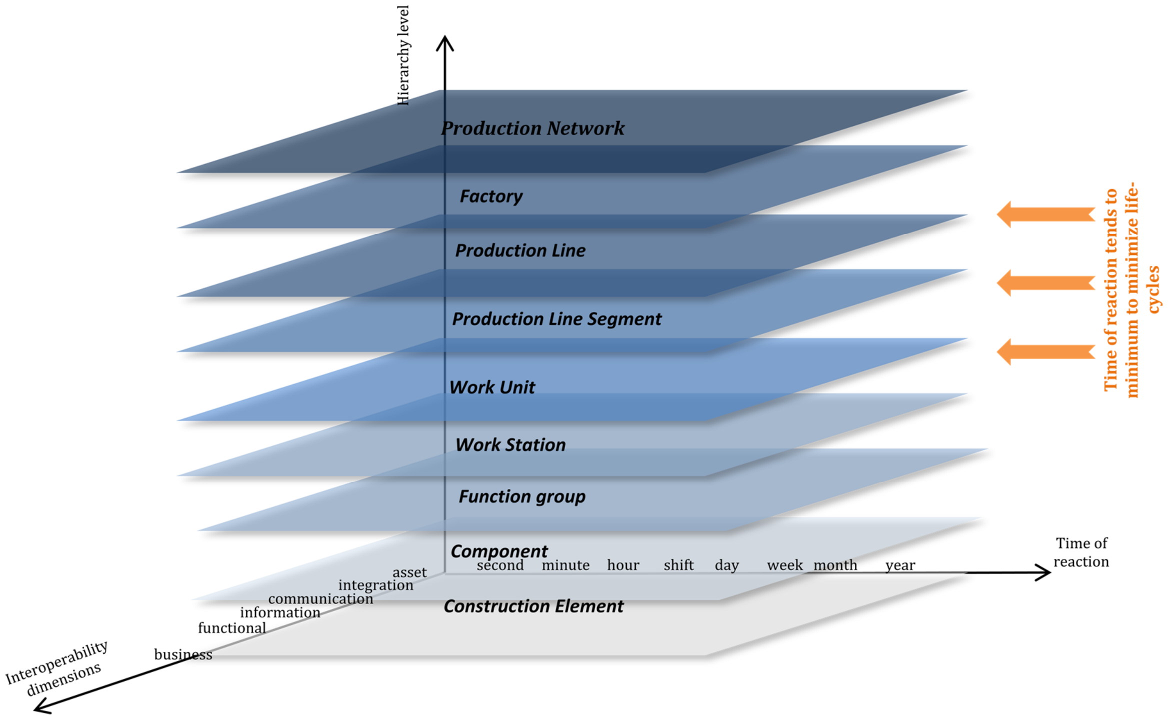


Figure 2.21 Automation pyramid considering information distribution

For each component,as it is based on the RAMI 4.0, information from every asset is distributed to all levels above. Information relevance is not easy to estimate for each level, for some reasons the data from sensors can be more important at the moment, than information about every worker that depends on the factory size.But the amount of information is easy to estimate. For the regular sized enterprise the biggest role plays information exchange between ERP and MES (and it is noted on Fig.2.16), it is more complex, that among sensors and actuators, the amount is larger and consists of resource information (personal, equipment, materials), operations capability information, operations definition information, operation schedule, operation performance, manufacturing control information.

This is why the color of each level on the figure is changing from transparent to saturated showing the amount of information. But in 5 years after digitalization of horizontal and vertical value chain, information will be interconnected within the whole architecture and will be equivalent important.

So, in future, next wave of innovation, providing smart objects, cyber-physical systems and Internet of things (IOT) [IOT16] will dissolve the automation pyramid on all, from ERP level to field control levels into “smart” objects in the cloud (Fig. 2.22). More information is in the book [Kue15].

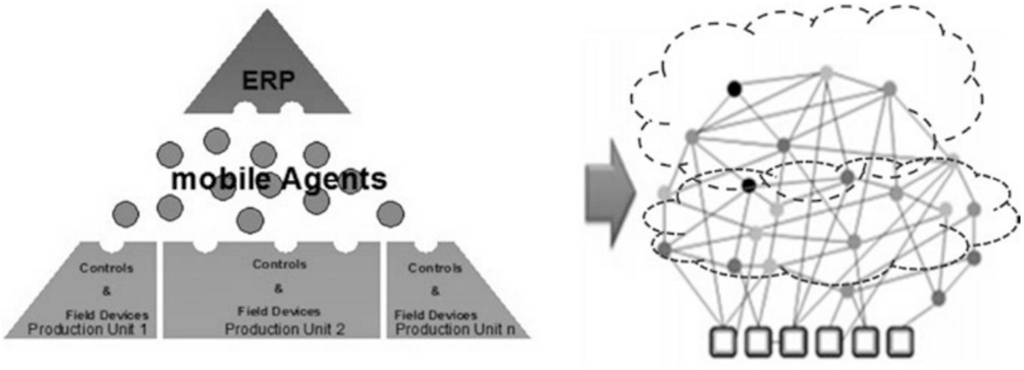


Figure 2.22 Dissolution of automation pyramid with agents into CPS with hybrid cloud according to [Kue15].

The enterprise control system would comprise all automation pyramid levels and all information distribution layers. But as a smaller control systemis going to be presented in this thesis (a technological process control system), the future architecture will be located on lowest levels of automation pyramid according to [ISA95] and particularly will include information from highest levels, ERP and MES. In this way question number 3 is particularly answered, the other part of developing steps is presented in next chapter.

3. Development of the control system

This chapter, as the previous one, is general and continues with naming fundamentals. In this chapter, all factors related to the control system development will be considered.In the beginning, the developing phases advised by VDI standards are considered to understand the application of its steps, and to estimate whereflexibility can play a significant role in answering the influences. In the end, the requirements for the control system development, such as software and functional, will be given.

3.1 Approaching the architecture design

In the last chapter all factors related to the control system development were discussed. Beginning with the factors, named “triggers”, for changeable production, the horizontal and vertical value chains supporting the product life-cycle, continuing with the understanding of what flexibility of production is and how to answer on triggers and disruptions, and ending with definition which features the flexible control system must achieve to be I4.0- and CPS conformed. But the main question how to develop flexible control system, and the possible positioning of it, will it be in the end an intelligent system,are still not answered.

At first, for the successful development and integration of the control system into the compound enterprise, the over the years proven industrial standardsof system development will be used.

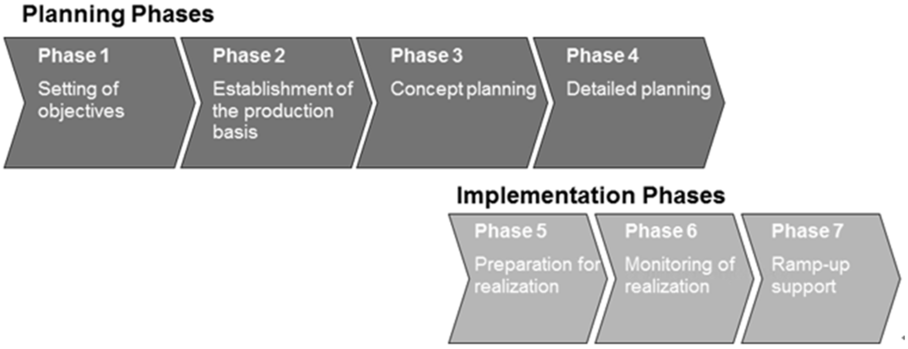


Figure 3.1 Plant planning phases due to [VDI5200]

NOTE: in this chapter, the term Product is considered as “thing that is the result of an action or process” according to Oxford dictionary [OxD16]. Corresponding to this definition, the term “Product”can be referred both to the control system or production system, which occurs as the result of the development process, oralso the detail, obtained as the result of the production process. In this thesis, by the term Product,a production control system, which appeared as the result of the development process, is meant.

Due to that, any technical product development, even a small detail or a production system or a big plant, can be described as a chain of activities that outlines the developing steps from generating the idea of the product till the product implementation into the particular process. In the case, the “product” role can play everything that was developed even till the big enterprise, because a product is a good or idea obtained in the result of development or production process. In the case of the thesis a product is the obtained control system.

For example, the VDI 5200 [VDI5200] guideline describes two phases of the plant planning: planning and implementation (Fig.3.1), which are the parts of the life cycle. And as described in the previous chapter, all assets (products) have similar life cycles according to I4.0.

To define the practical steps in the direction of control system development, also, the standards VDI 3695, VDI 2243 andVDI 2206 [VDI3695,VDI2206, VDI2243], are considered and compared. The development stages, as the part of life-cycles of different products, form plant till small detail are mostly the same. At first, the idea should reveal itself, a market should be analyzed, and then the product should be developed, tested and analyzed before implementation.

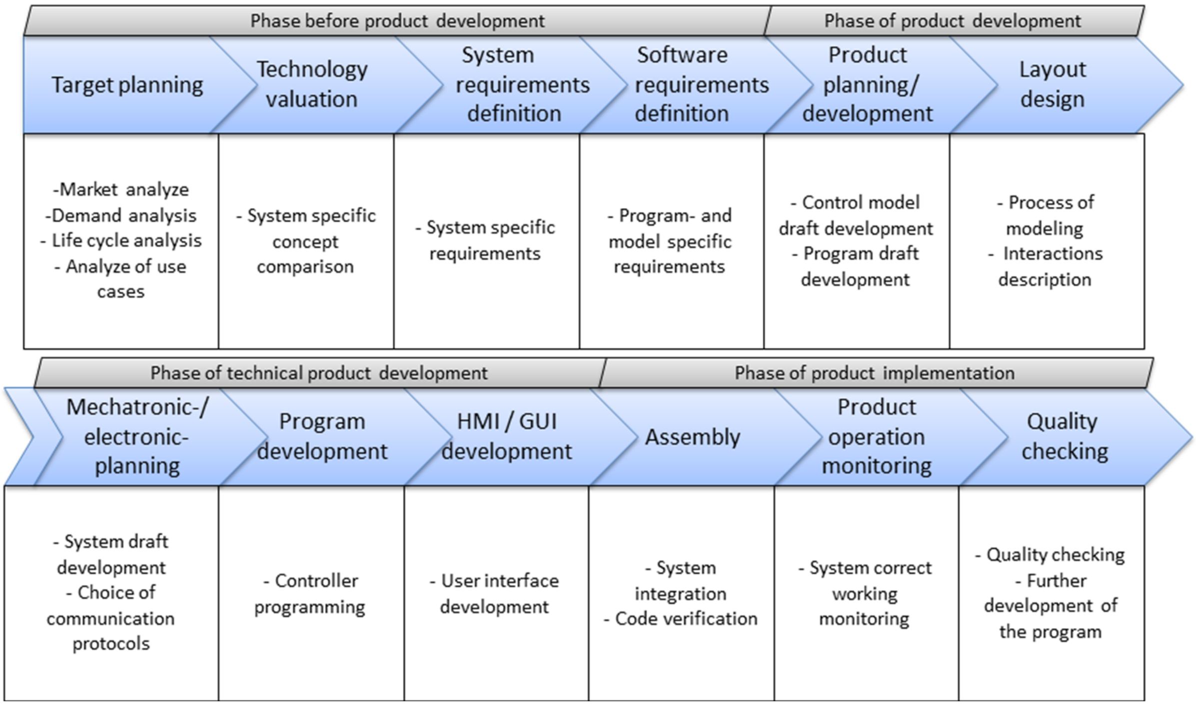


Figure 3.2 Control system planning phases

On Figure 3.2 the general stages of the control system development as a new product are presented, that were obtained from the analysis of the main points of different standards and product development stages proposedonthe Internet [Dem12].

Combining the known data about the development stages, concerning the control system development the phases look like as following (Fig.3.2). The planning can be separated into 4 phases: phase before product development, phase of product development, phase of technical product development and phase of implementation of the program.

Phase before product development includes market analysis and requirements specification. Phase of product development consists of a model draft and layout design. Phase of technical product development consists of mechatronic and programming planning, also HMI development and system integration. Phase of implementation as the last stage is the implementation and monitoring.

After steps definition, the design of the new control system will be starteddirectly, executingthe order fromthesteps in the given order according to Fig.3.2.

3.2 Phase before product development. Target planning

In this paragraph “Phase before product development” (Fig.3.2) consists of 4 steps: target planning, technology evaluation, system requirement and software requirements definitions. At first, a couple of words to the “Target planning” should be said. Demand analysis from Chapter 1, 2 shows that rapidly developing markets need new approaches to adapt fast to changing environment. Use cases for different domains of production are fully described in Chapter 1 (Table 1.1): a heater (heat production), a conveyor (logistics as part of production) and an assembly robot (assembling various parts is production). In the next paragraphs, the answer to question 1will be presented.

3.3 Technology valuation – intelligent systems

3.3.1 Artificial intelligence

In this paragraph will be defined the intelligence of the control system, will be presented different factory control approaches, e.g. knowledge-based technologies in common, supervision in systems, agents as one of knowledge-based engineering approach and multi-agent systems, as a set of cooperating agents, will be discussed.

During the operation, the production system has to stay stable and need to be flexible to cope with a dynamic environment. To automate the consideration of the occurring changes, avoiding constant human helpand costs, intellectualcontrol approaches should be taken into account. In this chapter the answer on the Question 1 is discussed, how to provide an intellectual filling of the flexible control system. In such systems to provide the flexibility, the control approaches based on artificial intelligence (AI), supervision, on experts´ knowledge are used. During the advantages analyze will be clear that intelligent systems are able to prepare production systems for the challenges of the future, due to that fact that they are considered as the production systems whose various control layers from shop floor to business level are enhanced with knowledge-based technologies and engineering.

On Figure 3.3 there are different types of artificial intelligence presented and collected due to literature analysis [Bub16, Rus95], they are applied in the different domain and interrelated during usage. So, for example, during the production optimization self-adaptation of the control rules can be realized by learning fuzzy logic through genetic algorithm [Nan12] or even fuzzy logic with diagnostic agents [Gen06].

According to the book [Rus95], the first work about AI was made by McCulloch and Pits in 1943, where they proposed a model of artificial neurons. In 1956 McCarty presented new field “Artificial intelligence”. Afterwards as a continuation of the work the genetic algorithms appear in 1959, which were based on the mutations of machine code, afterwards appear expert systems (70ss), Parallel Distributed Processing (by Rumelhart and McClelland, 1986) and intelligent agents and multi-agent systems in 90ss [Rus95]. Nowadays, for example, artificial intelligence is presented in robots performing human assistance.

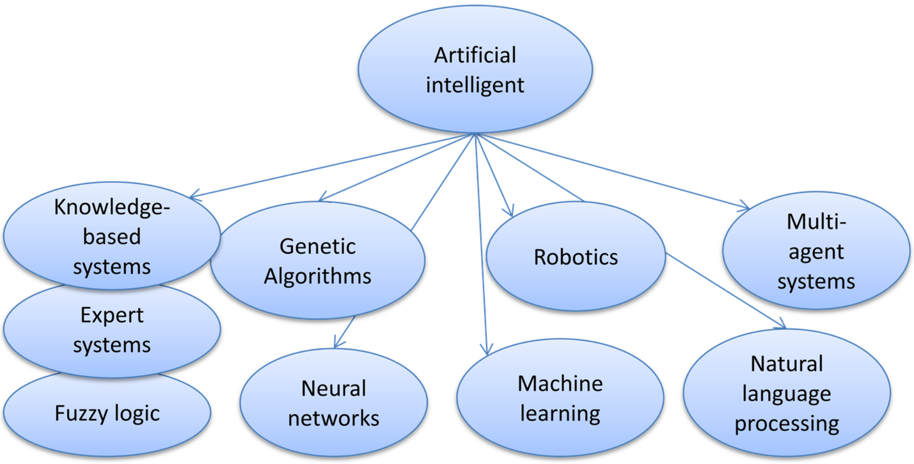


Figure 3.3 Particular representation of Artificial Intelligent manifold

Examples of flexible production systems that use the knowledge to support suchautomatic decisions have been promoted in research agendas and approaches [MAN16, Industrie 4.0, Lei09]. Firstly, the knowledge-basedsystems will be discussed.

3.3.2 Knowledge-based systems

Knowledge in knowledge-based systems consists of facts, theories, rules and procedures. According to Tasso et al., the Knowledge-based system (KBS) is a software system, which is capable to support the explicit representation of knowledge in some particular competence domains and of exploiting it through appropriate reasoning mechanisms to provide high-level problem-solving performance [Tas98]. As the KBSs represent artificial intelligence and are capable of performing the functions, which are traditionally considered as unique and exclusive of human cognition, KBSsare autonomously capable in problem-solving. Knowledge representation can be considered as the classical mathematical model extension in traditional control systems [Bub16]. Knowledge-based systems have been used to perform complex engineering tasks, in the control of complex technological and manufacturing processes, in flexible manufacturing systems with robotics and manipulators, in design, diagnosis, data interpretation, scheduling, planning, monitoring and so on [Tas98, Tza89, Bub16].

A representative of KBS are the expert systems, they perform intellectually demanding tasks at expert level performance, that they emphasize domain specific methods of problem-solving over general algorithms of computer science, also, they provide explanations for the reached conclusions [Tza89]. They are often used in different domains besides engineering – also, in medicine and business. Expert systems are the systems based on fuzzy logic [Fin14] and could be mixed with other artificial intelligence methods (e.g., Neural Networks, Genetic Algorithms, etc.) to reach better performance [Mar12, Nan12] (Fig.3.1).

Advantages and disadvantages of such systems are discussed in the literature [Leg13, Leg11,Tza89]:

- Knowledge-based (KB) technologies offer benefits for detecting disturbances since these tasks rely on a vast amount of knowledge and for their compensation.
- These systems implement the control using the control rules.
- They enforce a homogeneous representation of knowledge, allow incremental knowledge growth through the addition of rules and allow unplanned but useful interactions (also due to knowledge base´s proactivity operator can lose the control).
- As positives Legat et al.name in [Leg11] following, it can improve the engineering by reducing effort, and the provision of adequate, explicit formal knowledge

models for operating knowledge-based systems addressing future production system challenges.

As the drawbacks [Tza89] names points:

- The knowledge-based systems are complicated to build, due to the expert´s knowledge quantity.
- The control possibilities should be explicitly enumerated, and they have almost no capability of system generalization.

The general idea of the knowledge-based control is that control algorithm generates control decisions based on the knowledge description and some external information containing the requirements concerning the plant and the purpose of decision making [Bub16]. Knowledge usually cooperates with the traditional database containing the specific knowledge of the specific plant. There two types of the expert knowledge used in the system: knowledge of the plant (a descriptive approach) and knowledge of the control (a prescriptive character). The learning process of the KBS may be considered as an adaptation in traditional control systems. For example in the heater, the performance is described with different states with special control parameter for each state. Based on the knowledge of experts (a prescriptive character of knowledge), the control system can be learned to perform in different states with better productivity. They don’t replace experts, but make the knowledge more available.

Supervision in such systems can be used for the improvement of control methods and can be realized in different ways. It will be discussed in the next paragraph.

3.3.3 Supervision technologies

Supervision technologies first appeared in the paper of Ramadge and Wonham [Ram87] like the idea of supervision in discrete event dynamic systems and development of supervisory control theory (SCT). Later this trend developed itself and gained momentum. Supervision is used for the adaptation of the controller, e.g. the adjustment of the control rules at runtime to the environment, which is characterized by frequent changes in the product order, production modes and equipment configuration [San96].

The supervisor may be used to ensure a proper behavior of the whole system and enterprise against general system conditions and for the adaptation of the controller. As the example of the case, the adjustment of the control rules at runtime, which is characterized by frequent changes in the product instructions, production modes and equipment configuration, could be named [SIE15]. An important feature of supervision is that it must deal with dynamic facts. Thus, the expert system will possess temporal properties, it must have a memory to store recent events together with the operating circumstances related to their occurrence [Tza89]. For the heater application, an expert system can calculate optimization algorithm of a combustion process, for the conveyor- that could be optimization of technological/ process parameters (speed of the band). Due to its advantages, the supervisor feature will take place in the control architecture, so the opportunities to implement the supervisor in the standard closed- loop control on different levels of the automation pyramid are considered here.

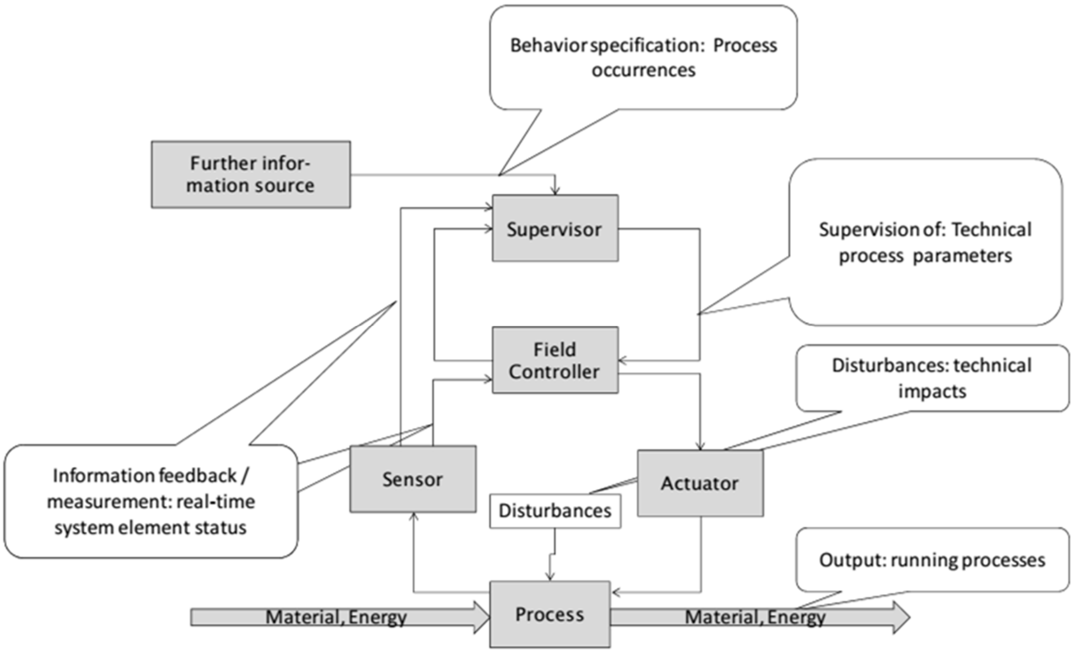


Figure 3.4 Supervisor impact at field control level according to [SIE15]

Supervision can be presented over a running controlled process on the field controlor even higher, following the automation pyramid, taking into account different data sources: sensors, actuators, process and process disturbances, performing changing the technical process parameters and also interacting with ERP and MES [SIE15]. The different cases of supervision were discussed in the project AMProC and presented on Fig. 3.4-3.6. From the figures, it is evident that depending on a hierarchy level, the supervisor behavior specification, the information feedback to a supervisor, the object of supervision and also the type of disturbances differs.

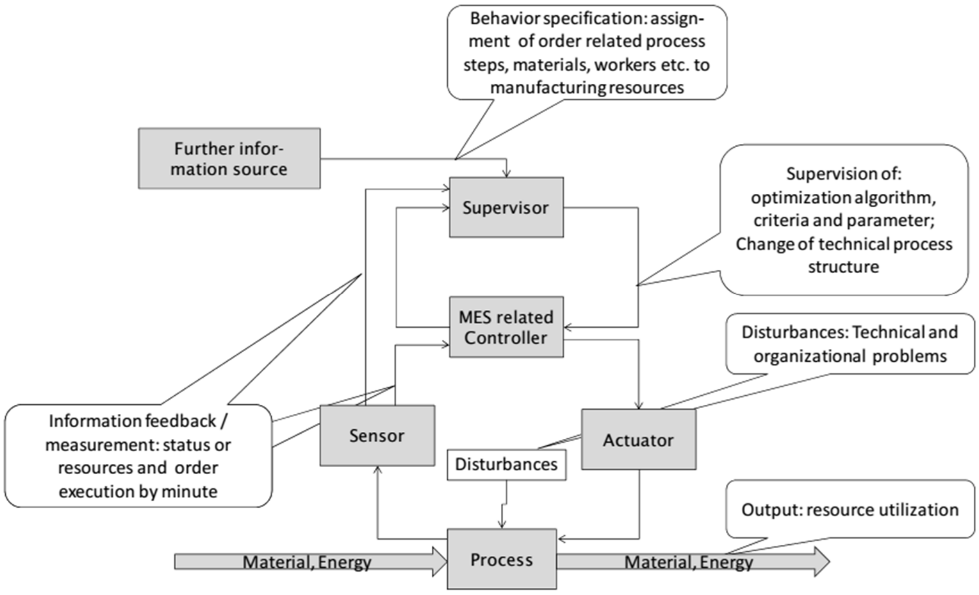


Figure 3.5 Supervisor impact at MES level according to [SIE15]

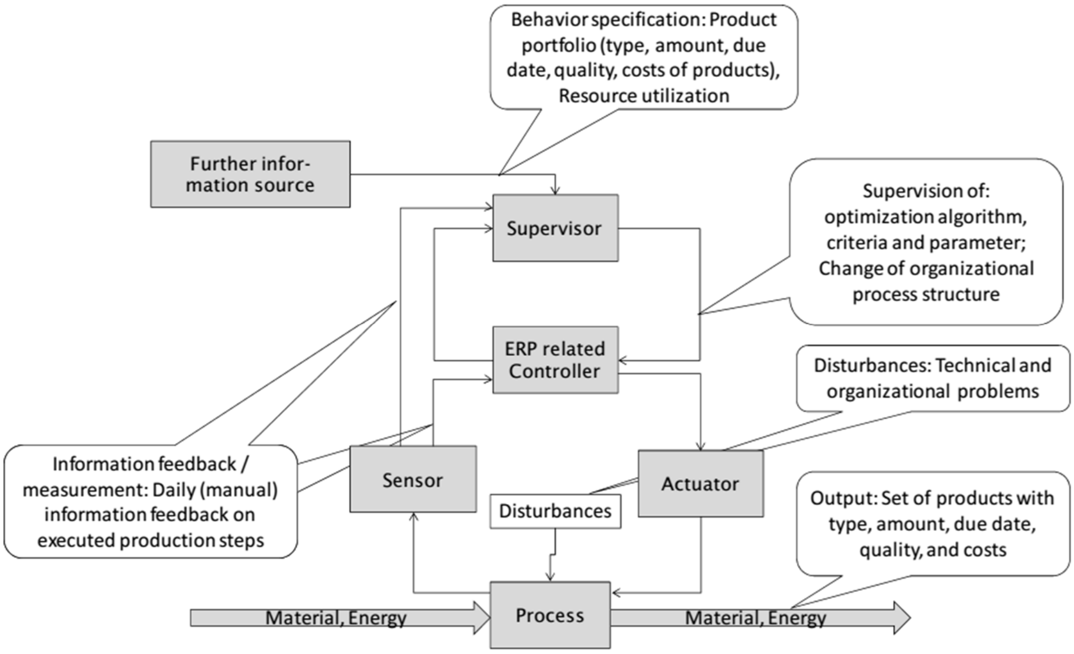


Figure 3.6 Supervisor impact at ERP level according to [SIE15]

For example, at the field control level, the supervisor controls technical process parameters receiving information about system elements in real-time from sensors and actuators placed on the shop floor. On MES level supervisor controls the process optimization algorithm and change technical process parameters. On this level, supervisor work is based in its turn on another type of disturbances: technological and organizational problems, and on this level supervisor collects more information about the system such as assignment of order related process steps, materials, workers, etc.

The same is valid on the highest level of automation pyramid – supervisor will control optimization algorithm, criteria and control parameters, also the organization of process structure. These tasks for the supervisor in our control architecture will be considered later in the final control architecture.

Talking about supervisors tasks, Tzafestas in [Tza89] claims that supervisor must handle qualitative and quantitative information. Supervisor (Fig.3.7) leads the discussion with a controller-process loop (this level supplies input-output and state variables) and information generator (provides the supervisor with useful information). Supervisor can detect, localize, classify, evaluate, make a decision and plan the action, and transfer this information to the operator based on that information of the controller supervisor.

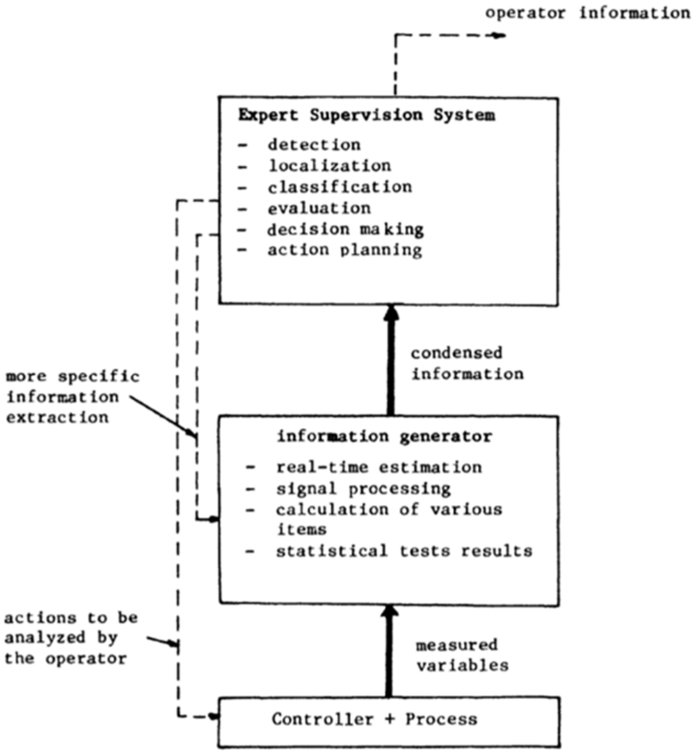


Figure 3.7 Supervisory control scheme due toTzafestas in [Tza89]

Summing up, supervisor functions in the flexible control system could be:

- dealing with dynamic facts, optimizing technological process (e.g. belt speed or acceleration [Kan14] and optimized control parameters for a heater[Rya15]),
- evaluation, classification and decision making in a process,interactionwith a process model in the direction adaptation and adjustment of control rules(e.g. [GRA16]),

- order-related communication with ERP/MES (e.g. in the directions bottom-up – “order to ERP system, like buy a new table” - for the conveyor case, top-down -“order to do a customized detail” – for the robot case). Orders appear as materials, quantity, cost resource utilization(service description), workers,
- adjustment of equipment configurations.

Later in Paragraph 3.3.6 different control approaches will be considered for thesupervision application inthe control system, now the next artificial intelligence type will be presented – agents.

3.3.4 Multi-agent systems

Due to the fact that agents is one of the artificial intelligence tool to and can be also used to construct flexible intelligent control architecture here the benefit of agents shall be described next.

An agentis an autonomous, problem-solving, and goal-driven computational entity with social abilities that is capable of effective, may be even proactive, behavior in an open and dynamic environment in the sense that it is observing and acting upon it in order to achieve its goals [Jen95].In other words, according to [VDI2653-1] agents represent a modeling approach encapsulated in software.

Agents possess following properties listed in Table 3.1 according to [Kar15a]:

Table 3.1 Agent properties

Properties of Agents	
Autonomy	An agent operates without direct intervention of human beings or other external entities
Responsiveness	An agent is equipped with sensors and actuators to directly interface with system environment
Proactiveness	An agent may proactively anticipate possible environments changes and react to them
Goal-orientation	An agent may take initiative whenever there is an opportunity to work towards its goals
Smart behavior	An agent has the knowledge of dealing with and solving problems in a specific area
Social ability	An agent interacts with human or other agents in pursuit of its goals
Learning capabilities	An agent needs to learn without intervention from the outside to be adaptive and autonomous

Usually, agents are divided into deliberative (proactive) and reactive agents[Kar15a]. Deliberative agents are comparatively flexible, but they can become too complex and slow in their reactions in performance. For the deliberative agent implementation, the Belief-Desire-Intention (BDI) architecture from Rao and Georgeffis distinguished [Rao92, Bus04].

Reactive agents have a simpler design because they don't have to deal with complex symbolic world model and respond quickly to relevant stimuli from its environment. Based on the input it produces the output by simple situation-action associations. In many cases, it is possible to replace a deliberative with several reactive agents without a loss of quality. Due to a simple implementation of reactive agents, sufficient for proposed control model, they can be used to construct the flexible architecture [Kar15a].

A collection of interacting agents is called a multi-agent system (MAS) [Bus04]. In a complex real-world the interacting cooperative agents can solve taskstogether. Through joined forces of semi-autonomous problem solvers, their communication, collaboration, negotiation and responsibility, they reach their individual goals.

MASgained fame due to the following features listed in Table 3.2:

Table 3.2 Multi-agent system properties according to [Kar15a]

Properties of MAS	
Decentralized control	MAS always have decentralized architecture and control
Flexibility	MAS can adaptto changing situations and requirements during execution
Adaptability/ reconfigurability	MAScan better fit execution plans referring the evolutionary nature
Scalability	MAS may run on a casual number of networked computers
Leanness	MAS are lean due to exactly coverage of clearly defined, limited field of expertise
Robustness/ fault tolerance	MAS have an ability to manage themselves autonomously even when problems or failures occur

According to many surveys about agents' application, different existing agent architectures and industrial application are known and will be described in next paragraph.

3.3.5 Agent architectures

In last decades several methodologies using agents such as holonic and multi-agent applications appeared [Chr94, Goe13]. Elaboration of a direction of agent researches was provided from the Rockwell Automation Inc., investing a large effort in researching into the alternative control solutions based on holonic and multi-agent systems [Vrb14].

Holonic and agent-based systems differ from traditional systems by using parallel distributed computation bringing benefits in reaction time to disturbances.

In the book [Bus04]Bussmannet al. describe Holonic manufacturing (from Greek “holos”), saying that they were developed from early 90ss to deal with increasing complexity, product customization, volatile demand and cost pressure. This architecture consists of autonomous entities, which operate in a flexible hierarchy.He claims that in comparison to heterarchical architecture, Holonic manufacturing takes a much larger perspective than heterarchical control architecture, looking at the whole manufacturing process, not only control. But in contrast to hierarchical control systems, holonic manufacturing systems (HMS) create loose and flexible communication, which never force a holon to perform a certain task [Bus04].

Agent-based control is a software technology, which provides autonomous and co-operative behavior. It can be realized hierarchically or heterarchically. Also, it can enable a realization of holonic architecture. Several holonic and multi-agent architectures were already proposed in science:

1. Reference architecture for holonic manufacturing systems PROSA (Product-Resource-Order-Staff-Architecture) which uses holons to represent products, resources, orders and logical activities and consists of basic holons: order holons, product holons and resource holons [Bru98]. PROSA cover hierarchical and heterarchical control approaches.

2. ADACOR (ADAptive holonic COntrolaRchitecture for distributed manufacturing systems) holonic architecture addresses the agile reaction to emergence and change, increasing the agility and flexibility of manufacturing control systems, located in volatile environments characterized by the frequent occurrence of disturbances [Lei05, Lei13, Bar15].

3. GRACE (InterGration of pRocess and quAlity Control using multi-agEnt technology) [Foe12, GRA16] multi-agent system – a distributed manufacturing control system for production lines producing washing machines [Rod13].

4. PABADIS (Plant Automation BAsed on DIstributed Systems) – uses the concept of co-operative manufacturing units to provide significant functions to the production process in automation control, encapsulating residential, products and shop floor management as agents. The approach consists of centralized (for the connection with ERP) and decentralized components, being the products implemented using the mobile agent technology. Building blocks in HMS are mostly agents and are placed in the central unit, collaborating remotely with the processing unit building blocks. In contrast to HMS, in PABADIS the order building blocks are mobile, migrating through the system and performing their tasks on different control entities depending on the used processing unit building block [Lei09,Lue04, Fer11].

There are hundreds of other architectures and others applications or methodologies oriented on an industrial domain are described in surveys [Lep15, Lei13, Lei09]. Dynamic hybrid control architectures in manufacturing control systems are compared in [Jim15].

MAS are not only an academic concept anymore.Different approaches in the direction of MAS application in the industry are known, e.g. suitable for chosen application cases. For heater application there are available: Multi-Agent Control System of a Kraft Recovery Boiler [Iva08], Multi-agent simulation in inference evaluation of steam boiler emission [Had05]. Agents’ usage in intelligent energy systems, related to Smart Grids, is presented in the review of Pavel Vrba et al. in [Vrb14]. MAS applications for Conveyor are: Holonic Chain Conveyor Control System [Bel09], production plant using agents as connectors between legacy systems and I4.0 cloud [Fau15],a shop-floor control system [Vis98] and the planning system of the plant resources [Bey15]. Based on these industrial applications, it is shown that it is valid to pursue an agent-based architecture in the thesis.

The following real running industrial implementations (mostly in Germany) are known: Daimler Chrysler´s Prototype P2000+ cylinder head manufacturing system for diesel engines near Stuttgart, Holomobiles – to optimize material flow and productivity in production units [Bus04], Provis.Agent monitors and controls the body, paint and assembly shops in DaimlerChrysler´s automotive plant in Bremen [Sau06], Agent-based chilled water system for US Navy ship from Rockwell Automation, etc. More information about applications can be found in [Lei09, Vrb13].

All surveys always are in favor of the use of agents, because MAS, as intelligent and decentralized control systems, have an indispensable role to play in enabling the overall resilience of the combined cyber-physical engineering system claims Amro M. Farid in [Far15]. Also, agent-based industrial systems have many named before agent benefits: robustness, scalability, reconfigurability and productivity [Lei13]. Although the agent technology keeps proving its benefits over traditional approaches, there is no massive application in industry. The reasons for that are technology factor and human factor [Vrb14, Sei13]: at first, operators and engineers are not skilled to use new technologies; second, there are some bottlenecks from the industry side, namely, the investment, interoperability, real-time constraints, etc. Despite this, agent technology has the potential to meet the future challenges in production control, because it provides conceptual models and implementation architectures for goal-based decision making, negotiation and coordination, which will be indispensable features of future control systems [Bus04].

Comparing MAS and traditional manufacturing control systems, Paulo Leitão in the survey [Lei09] names the time problem solving of the modern manufacturing control systems, which are traditionally implemented using centralized and hierarchical control approaches (more detailed in next paragraph). They possess following functions: planning, scheduling, execution (dispatching, monitoring, etc.) (Fig.3.8). Control algorithms are complex, with a variety of executed functions. Scheduling and planning have not the same execution time plan that causes time problems especially when the detailed up-to-minute knowledge about the process is needed. This requires a control software which acts dynamically and adapts to changing order requirements [VDI2653-1].In his survey, he claims that intelligent and distributed manufacturing control systems are required to fulfill the gap left by the traditional centralized approaches.

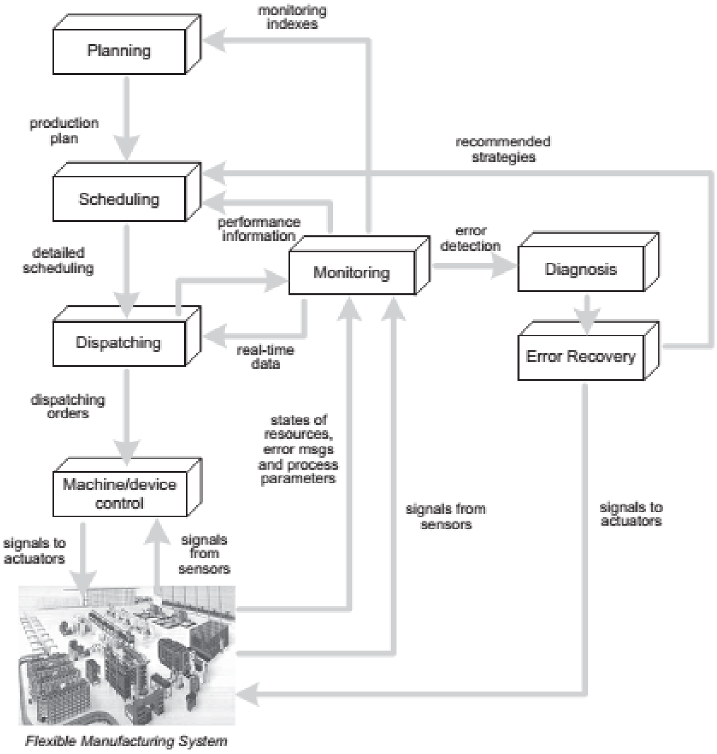


Figure 3.8 Traditional control of manufacturing systems according to [Lei09]

He says, MAS control systems address intelligent and distributed control and they are characterized by following features according to [Lei09]:

- In the distributed approach, a complex problem is divided into several small problems, mapped on intelligent control unit;
- Each control unit is autonomous having its own objectives and knowledge, and encapsulating intelligent functions;
- The global control decisions (e.g. the scheduling, monitoring and diagnosis) are determined by more than one control unit, i.e. the control units work together, interacting in a collaborative way to reach a production decision;

- Some control units are connected to physical automation devices, such as robots and CNC machines;
- Control units should exhibit several important features such as re-configurability, robustness and learning.

Also, he presented a comparison of the traditional control solutions and distributed intelligent solution in Table 3.3.

It is worth recalling that ideas of Industrie 4.0 requires the application of an easier processing of data and also interconnection of all automation pyramid levels, using internet technologies that allowing new methods of data integration and data analysis. It is important to be aware of all system resources for the intended use of them to produce customized products. So, systems based on agents are the easy and not expensive way to apply I4.0 and also to allow the migration from legacy systems to the I4.0 [Ros15].

Table 3.3 Comparison of the traditional control solutions and distributed intelligent solution according to [Lei09]

Traditional control solution (e.g.CIM)	Distributed intelligent control solution (e.g. agent-based)
Centralized solution for each individual control function	Distributed solution with cooperation between nodes and focusing on more than one control function
Rigid and static architecture	Flexible, programmable and dynamic architecture
Communications one for many (1-N)	Communication many for many (N-M)
Intelligence centered in the top levels	Intelligence distributed by control levels
Efficiency through the specialization	Efficiency through the flexibility
Weak response to disturbances	High response to disturbances
Operators are replaced by automation technologies (removed from production process)	Operators are complemented with automation technologies (increasing the skills of the operators that stay in the production process)

Using agents all elements in the system will have the same communication standard to allow the data- and function- exchange in the whole system. So, every element is defined, and all his features and abilities are initially known. Besides, agents possess many features allowing implementing I.4.0. [Bau14].

After this literature analyze, the use of MAS for the flexible control system is justified. Now, in Paragraph 3.3.6 different control approaches will be considered for the cooperation representation of MAS and supervisor.

3.3.6 Choice of control system architecture

Different production control architectures have been developed with time (Fig.3.9). In the beginning, an easiest one - centralized architecture- appeared, later, more complex distributed architectures maintained essential similarities appeared [Die11, Lei04].

Usually, control systems have the similar structure consisting of Control applications (C), control Devices (D), sensors and actors,which manage and impact Processes (P), which is apparent in every production control system (Fig.3.9). On Figure 3.9 the collection of existed control structures from the papers [Die11, Dil91, Lei04, Bus04] are presented.

a. Mostly in production the traditional control systems, centralized and hierarchical architecture can be meet (a, b) [Vrb13]. Traditional centralized architecture has a one main controller over the devices to control a production process, for example, the controller of the water heating. This type of the controller prioritizes process robustness.

Other architectures have more than one controller, so it is assumed to call them decentralized.

b. Hierarchical architecture presents a distribution of control tasks and decision-making between multiple controllers, where a tree-structure provides a master-slave relation between controllers. The main advantages of this architecture are robustness and predictability, also the efficiency of this approach here is better, than in centralized approach [Lei04].

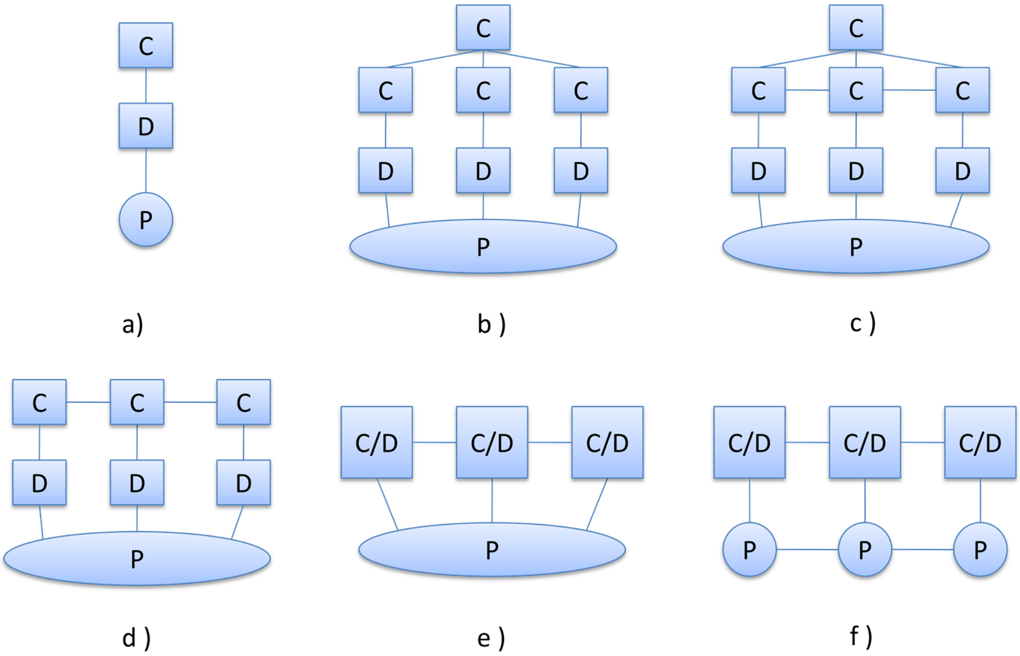


Figure 3.9Diversity of control structures: a) centralized,b) hierarchical, c) modified hierarchical, d) heterarchical, e) distributed, f) mechatronic according to [Die11, Lei04].

Despite the frequent use, traditional methods don’t show the capability of adaptation: centralized and hierarchical control approaches present proper production optimization, but a weak response to change, because of rigidity and centralization of the control structure, also the appearance of disturbances reduce system performance. Hierarchical and centralized systems are difficult to modify of extending and they exhibit a very low fault tolerance [Bus04].

c. The modified hierarchical approach (c) tries to improve the response to disturbancesadding the interaction between controllers, maintaining all the features of a hierarchical approach. Due to the interaction at the same control level, the expandability of the system is easier than the hierarchical architecture [Lei04].

d. Heterarchical production control methods (*d*) have the equal interaction between all controllers. They present a good response to change, but due to partial knowledge about the system optimization may be degraded [Lei04]. Heterarchical control increases the autonomy of each controller, while in hierarchical architecture it is hidden under supervision [Lei05].

e. Distributed and heterarchical architectures (d and e) are almost the same, but the distributed architecture integrates together the device and control methods in place [Die11]. Distributed architecture makes automation solutions more flexible, in these architectures, there is no hierarchy in the control.

f. The mechatronic structure (f) differs from other architectures. It connects the complexity reduction of decentralized and distributed system with a good analytic ability of the behavior description of centralized control architecture and allows the efficient developing process of a production system, besides this architecture is self-containment [Die11].

Recalling the challenge to develop a production control system with autonomy and intelligence capabilities (that features agents already have), it should be agile, and fast in adaptation, expandable, but also robust, integrated into legacy systems the architecture with the same characteristics should be chosen.Heterarchical control approaches introduce a good response to the flexibility and agility requirements. By adding the supervisor entity for the ensuring a proper system behavior, as was described in the paragraph about supervision, a hierarchy in a decentralized system is introduced. Robustness will be taken from mechatronic architecture, which saved the features of centralized architecture. In this way, the choice fell on a mix of modified hierarchical architecture for the MAS realization. The control problem in the mixwill be decomposed in several simpler problems and distributed among multiple control-layers, controlling at the same time different processes, which are supervised. The levels in the mix are distributed as a tree structure, allowing the distribution of decision making among the hierarchy. In this way, all benefits of mechatronic and modified-hierarchical architectures are merged.

Answering the question about control system positioning (Question 3), other scientists propose the ideas of using MAS on different levels of automation pyramid, supervision usage for particular objectives are known. E.g. the idea of supervisor and MAS cooperation on PLC is described in [Pri14]. Hybrid multi-agent approaches, where a part of MAS is placed in MES level and other part on the field control level and its considered integration are presented in [Ule12, Ule14].

The MAS architecturein the thesis, how it was mentioned before after automation pyramid analyze, will be located on the field control, above the legacy systems and particularly will include information from ERP and MES. For its integration will be no reconfiguration needed, it exists autonomously, without integration into ERP or MES, but communicating with them.

Respecting the standardVDI2653-1[VDI2653n], the architecture will be constructed using JADE platform in Eclipse[Ecl16] and later will be evaluated once again all the advantages and disadvantages of the development. For the agents´ definitions and interaction between them the DACS methodology will be used. Both JADE platform and DACS methodology are described more detailed in next two paragraphs.

3.3.7 DACS

VDI/VDE 2653-Blatt 2 [VDI2653-2] contains a collection of agents systems design approaches, such as SODA, AgentUML and others, which provide a set of models, engineering steps to organize a running agent system [Kar15]. The book about agents systems [Kar15] claims that DACS method remains the most abstract and adequate methodology, related to the engineering of distributed control system.

A design methodology for agent-based production control systems should obviously provide models and methods that capture key agent-oriented aspects of the design to specify the agent solution [Bus04]. DACS methodology is able to model the agent-oriented aspects of a production control system, enabling the consideration of necessary control decisions independently, receiving in the end optimized decision structure[Kar15].

DACS methodology consists of following steps due to [Kar15, Bus04]. At first, we identify the *production control problem*. The specification of the production control problem consists of following steps:

- a specification of the physical production process to be controlled,
- a specification of the production operation conditions,
- a specification of the production goals and requirements.

According to the DACS methodology, secondly, the *control decisions*those are necessary to operate the production processes due to control problem, which of them are mandatory for the correct function,are defined. Also, the *dependencies* among control decisions are identified. Thirdly, *identification of agents* of the control system and the decisions they are responsible for are defined. The control decisions should be grouped and mapped to agents.Then, the *interaction*s between agents are specified. Afterwards, the agent model can be implemented. The implementation will be provided by JADE in this thesis.

3.3.8 JADE

The Foundation for Intelligent Physical Agents' (FIPA) [Bel07] organization produced sets of standards for agent management, agent communication and messages exchange. One of the most popular open-source frameworks in this field is JADE (Java Agent DEvelopment framework), that is a Java-based tool conform to the FIPA. JADE platform is a software agent system, supporting programming and running of agents application [FIP16, Bel07]. FIPA standardization efforts were aimed namely at software agents as autonomous entities communicating by using a specific agent language (ACL), and operating on a higher decision-making level or high-level control [Lei13].

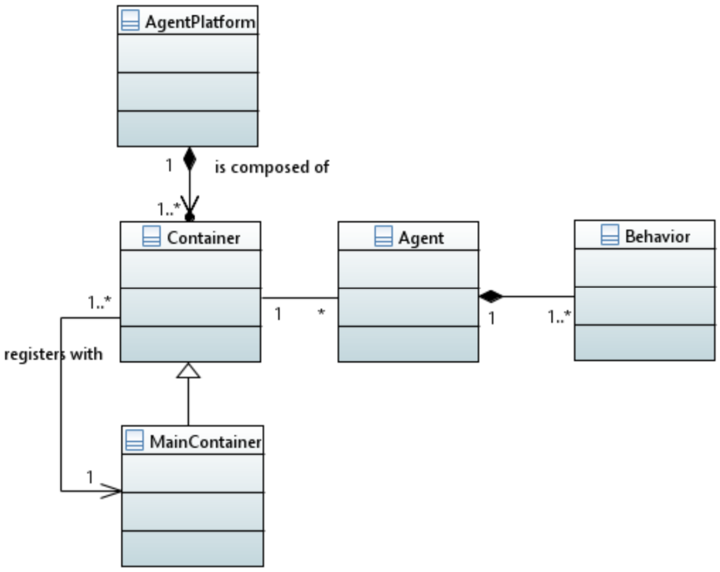


Figure 3.10 Relationship between the main architectural elements

JADE offers a filed communication mechanisms between agents [VDI2653n].All they are registered in Container in Agent Platform, so they have access to other behaviors. The realization of the Agent Behaviors using Activities takes place according to the Fig.3.10, where Class Diagram in SYSML shows the functioning of behaviors. The idea is taken out of [Bel07] with adding the dependencies between behaviors of agents.

Implementation of flexible MAS using JADE will be described in next chapter and now will be presented the next step of the control system development referring to Fig. 3.2 – the system and software requirements definition.

3.4 System requirements definition

Many experts claim that future industry will consist of objects with artificial intelligence, which will communicate, exchange information, negotiate and take decisions and, therefore, control themselves [Kue15]. Here, system specific requirements and system properties will be defined, to design an intelligent flexible control system, also making manufacturing system more flexible and adaptable towards “smart factory” paradigm. Also, the before defined MAS architecture used for the realization of a flexible control system should meet system requirements listed in this paragraph.

Many authors present the requirements to modern enterprises and manufacturing systems so that it can meet the challenges of automation [Vog14, Leg14, Bus99, Lue04, Lue10, Gir15]. First requirements which are coming from the management of the organization for the system development are: the control system needs to have a short time of design; low development costs and be also safe. Bussmann et al. [Bus99, Ant11] claim that to meet the challenges of modern enterprises, automation and control systems must be decentralized and product-/ resource based, control interactions should be abstract, generalized and flexible; control must be self-organizing.

Due to [Lue04] presented by Lüder et al.,it should meet the requirements of Flexibility on product volume and equipment changes;Human integration and friendliness;Compliance with existing system and standards;Integration with existing control devices and legacy systems;Extendibility and reconfigurability;Web Integration;Security, etc.

Giret and Trentesaux in [Gir15] present success of manufacturing system measured regarding flexibility, agility and versatility. They achieve it using distributed intelligent manufacturing system, using smaller manageable systems instead of a complex system. They claim that autonomy and co-operation are necessary to create flexible behavior and adapt to the changing production conditions.

They place following requirements on agile system:

- Manufacturing control systems require autonomous entities to be organized in hierarchical and heterarchical structures.
- Manufacturing control units require routine-based behavior which is both efficient and timely.
- Manufacturing control systems require standardized structures, standardized functional units that can be connected to the different levels in the system using standardized interfaces and communication protocols.
- Manufacturing control systems require sustainable production processes.

Also, Giret and Trentesauxsay that methodology for IMS should combine the entire range of manufacturing actions to model the agile manufacturing enterprise (order receiving, production, and marketing)[Gir15].

Antonova et al. claim in [Ant11] that to fulfill the requirements for agility, new generation control systems are challenged to solve the task ofautomation of the entire product life-cycle through full integration of information and control systems.

Table 3.4 MAS compliance to system requirements

Functional Requirements	Compliance
The control system must be decentralized with reactive entities	x
The control system must be self-organizable, component based and component aware	x
The control system should be extendable and reconfigurable	x
The control system should be flexible on product volume and equipment changes	x
The control system shall be product/resource based	x
The control system shall comprehend autonomous entities organized in hierarchical and heterarchical structures	x
The control system shall have efficient and timely routine-based behavior	x
The control system shall have standardized functional units connected to the different levels of the system using standardized interfaces and communication protocols	x
The control system shall be integrated with existing control devices and legacy systems	x
The control system shall be compliant with existing system standards	x
The control system should be human friendly	x

For distributed manufacturing the developments are anticipated by the introduction of the more specific Cyber-Physical Production Systems (CPPS) [Kue15], so the following system properties for the CPPS should be considered: modularity, heterogeneity, scalability, context awareness, autonomy, interoperability, networkability.

It is useful to remember, added by Broy key properties of the system to reach Cyber-physical-system-properties [Bro13] that were listed before in the paragraph of CPS description such as parallel data collection (via sensors), data fusion, processing of physical data from the environment, locally, globally and in real time, etc.

[Wey14] place requirements to allow a system to be Industry 4.0-appropriate: simple order award, automatic order planning and distribution between other units, all units may register themselves in the system, energy consumption must be considered.

Multi-agent systems due to their properties meet most of the listed requirements. Not to be unfounded, requirements have been formalized in Table 3.4 to show the MAS compliance.

After the control system properties and restrictions were discussed and it was proved which control architecture to use, requirements also on the control model are presented in the next paragraph.

3.5 Software requirements definition

For the software development requirement definition different researches were taken into account [Bus99, Ant11]. The basic requirements are listed in Table 3.5 with the MAS compliance.

Table 3.5 MAS compliance to software requirements

Software Requirements	Compliance
The control program shall have clear semantics	x
The control program shall provide data and process encapsulation	x
The control system shall lead straight-forward translation from the control task on a resource or function to autonomous entities, that can encapsulate and provide to consumers their functionalities and abilities as servers	x
The control system shall define a mixed top-down and bottom-up development process	x
The control system should combine the entire range of manufacturing actions to model the agile manufacturing enterprise (order receiving, production)	x

The set of functions of the control system was also analyzed. The functions due to migration from legacy systems to CPS and I4.0 in Chapter 2 were already defined (Table 2.2).

As an automated control system concerning defined production jobs, due to VDI/VDE 3694 [VDI3694], it should consider the following objectives of automation project:

- Concerning equipment: increased performance, improved quality, reduced emissions, reduced energy consumption, improved process management, improved reliability/availability, and improved safety.
- Concerning economic efficiency: rationalization and reduced costs.

Also, the following basic aspects have to be considered: process monitoring, process stabilization, process management, process optimization, process balancing (balancing of flows of material, energy and information), process safeguarding. All these functions will be considered on the stage of planning activities between the agents.

Later, these functions will be considered on the phase of control architecture selection.

Summing up the results of this chapter, a mixed control architecture from modified hierarchical and mechatronic will be received. Also,added supervisor will observe the work of the system. The flexible control system will be implemented as MAS, which is answering system requirements, and will have functions, which are answering software requirements.

4. Phase of product development

In Chapter 4, a generic architecture of the control system model based on multi-agent system which meets the requirement from the previous chapter is presented. For the modeling of the multi-agent system DACS methodology that was described in Chapter with fundamentals will be used. According to the methodology, in this Chapter necessary agents and their behaviors for three application cases will be defined.

4.1 Product Planning. Control model draft development

During the investigations provided in previous paragraphs, the rough model of control system was developed (Fig.4.1).

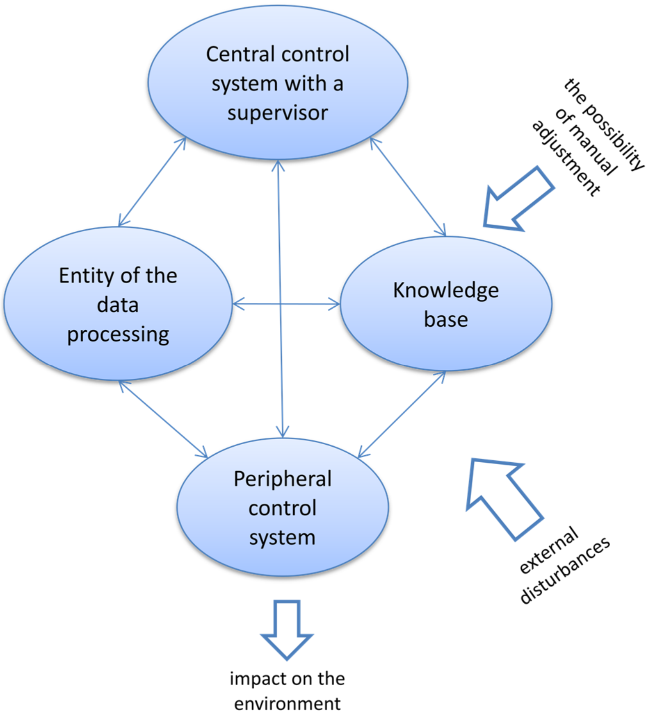


Figure 4.1 Rough control system model

The rough view of intelligent control architecture has a peripheral control system to proceed directly to field control, an entity of the data processing and central control system. The control system should have its own knowledge database and the possibility of the manual adjustment. Also, on the picture the external disturbances and impact on the environment are presented to show the close connection to the environment, to which it should adjust.

Supervisor performs tasks, described above:

- dealing with dynamic facts, optimizing a technological process ,
- evaluation, classification and decision making in a process, interaction with a process model in the direction adaptation and adjustment of control rules,
- order-related communication with ERP/MES,
- adjustment of equipment configurations.

Next,this architecture will be presented by means of multi-agents that shall be realized in JADE platform.The complex control task will be deployed between the agents to avoid heavily loaded agents through DACS methodology. The properties of each element of MAS will depend on the conditions defined by other elements. System properties are determined not only by the properties of each element but also their interaction properties.

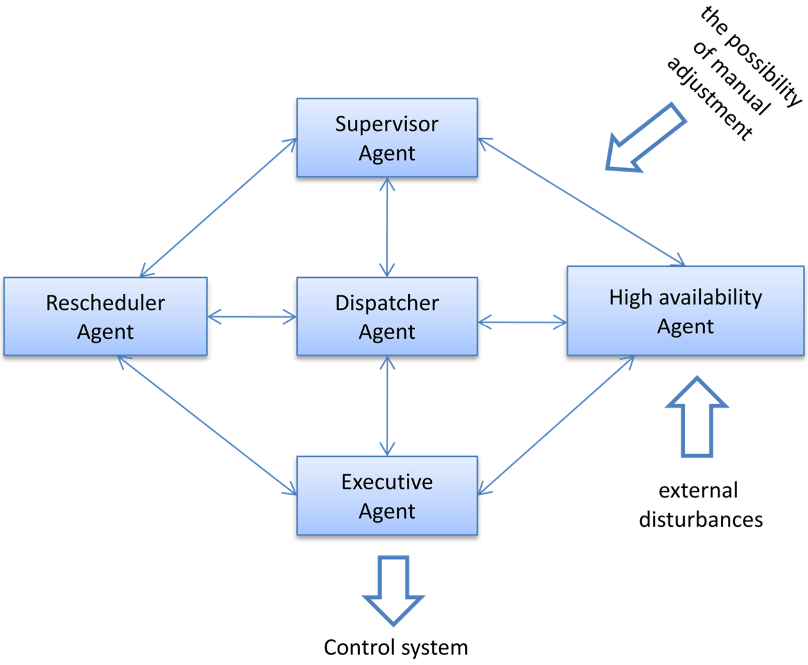


Figure 4.2 Rough agent architecture

For the four sets of control decisions identified in Fig.4.1 five types of agents shall be applied (Fig.4.2). These are:

- Executive Agent - applied to interact with legacy control systems,
- Supervisor Agent - applied to perform supervisory-tasks,
- High-availability Agent - applied to deal with external disturbances,
- Rescheduler Agent-applied to perform data processing within the process control,
- Dispatcher Agent -to deal with a knowledge base to ensure sustainable control.

In next chapter, the more detailed creation of layered architecturebased on agents with the presentation of particular tasks of agents will be shown.

4.2 “Conveyor” application case description with DACS methodology

As it was given before, DACS methodology consists of following steps: at first, *production control problem definition*is analyzed. Secondly, a definition of the *control decisions* and dependencies, and then *agents’identification and interactions* descriptions are defined. Steps will be performed in given order for the each application case, starting from the conveyor.

Specification of the production control problem

- a specification of the physical production process to be controlled.
- a specification of the production operation conditions.
- a specification of the production goals and requirements.

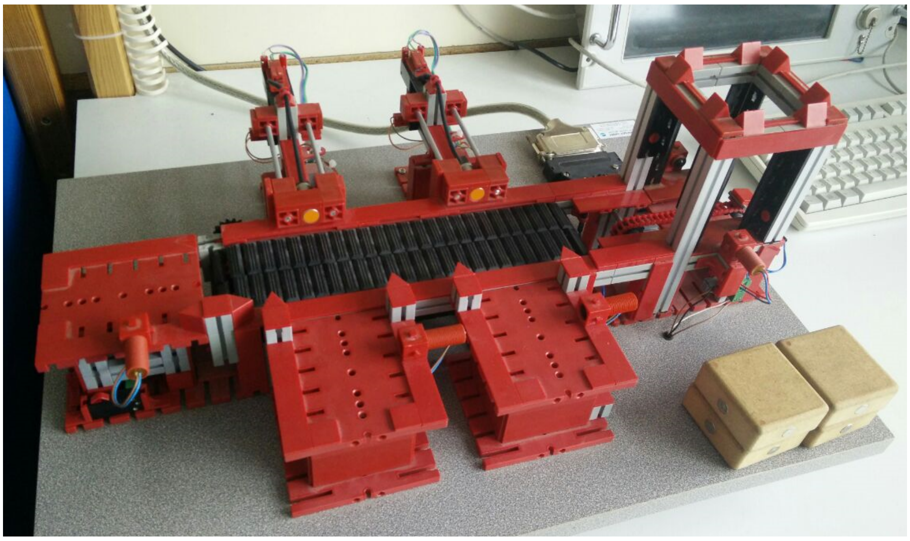


Figure 4.3.Fischertechnic model

The operational conveyor from Fischertechnik presented on Fig.4.3consists of a small band, big transporting band, 3 different destination- tables, 2 pushers, and sensors.

a) The **production process** is as follows: after loading station the coming workpiece that presents a transported package is identified and delivered to its destination depending on the product type. It will be transported to the certain table through the whole system on the band till it reaches the target place. Tables can play a role of the working stations processing the workpiece according to instructions, which will be also obtained in the identification stage. The system operating in this mode matches Industrie 4.0.

b) The **operational conditions** specify the input and output of the production process. Input is a “workpiece” that represents an order to get a customized product, in the case if the tables play roles of the workstations. Or, it is a package transported till the table. Also, the output of the production process on the Fischertechnik model is a product specified by order, which could be already customized to the desires of the client in the case if the workpiece will be changed on the workstations in the case of their existence. Or, it is a package arrived on the target place.

Disturbances may occur any time during the processing, such as unavailability of the resource: one of examples is a break of the table (or another working resource), another example is the destination place (table or workstation) is occupied.

c) The production **goal**: transporting with high productivity, with no failure. During the transportation optimization of the band speed to the type of the package (or workpiece) is possible.

d) Production **requirements**: control flexibility with respect to resource or order changes, Robustness with respect to failures, quality assurance, maintainability, existence of fallback strategies. These tasks will be considered further in the task table for different agents.

The conveyor control systemcanmeet these requirements, with opportunities to transport different workpieces to different tables, ensuring flexibility in path selection according to the identified element on the band. Thus, during the control an element should be identified and an appropriate control rule should be chosen. At the same time availability of conveyor parts (tables), providing a path for the element, is ensured.

Analysis of control decisions

According to [Bus04] DACS methodology says, the first step should identify any solution to the given problem, starting by looking at the physical actions that are necessary to run production process and identify those entire situations in which the controller has alternative actions to choose from, also to define the effects after chosen decisions tasks.

These decisions are called effectoric decisions and they are necessary to run the production process. Identification of effectoric decisions is facilitated through the analysis of typical scenarios occurring during the process is presented in Table 4.1.

Table 4.1 Scenarios of the conveyor

Use case	Scenario content
Pre-conditions	Workpiecearrived on the Loading station
Main scenario	1. Define the type of the workpiece 2. Ask the target table 3.1. If Workpiece was known before, then take the target table from the resource table 4. Check resource availability 5. If all resources are available, prescribe a free table to the workpiece 6. Transport to the target
Other scenarios	4.2. If the table is absent, make in order to ERP 4.3. Implement after it arrives 4.4. Add the table to the Resource table
Post-conditions	Workpiece is delivered to the target
Error-conditions	3.1. If the Workpiece wasn´t known before, then add to the table of targets 3.2. Prescribe the target table to the workpiece 4.1. If the table is not available check if it occupied. 4.2. If the table is occupied, then wait till its free

Dependencies between control decisions

- Transportation of the block to the target table is possible after block identification.
- Transportation of the block to the target table is possible if the resources to bring the block to target are available.

Trigger diagram considering agent scenarios and dependencies is presented in Fig. 4.4.

Identification of agents

Not to make it too complex for one agent, the control decisions should be suited to different agents, so that still the control problem will be solved, and modifications that improve the decision model are allowed. To perform the simple transportation to the needed table, the control system needs to know all information about available resources.

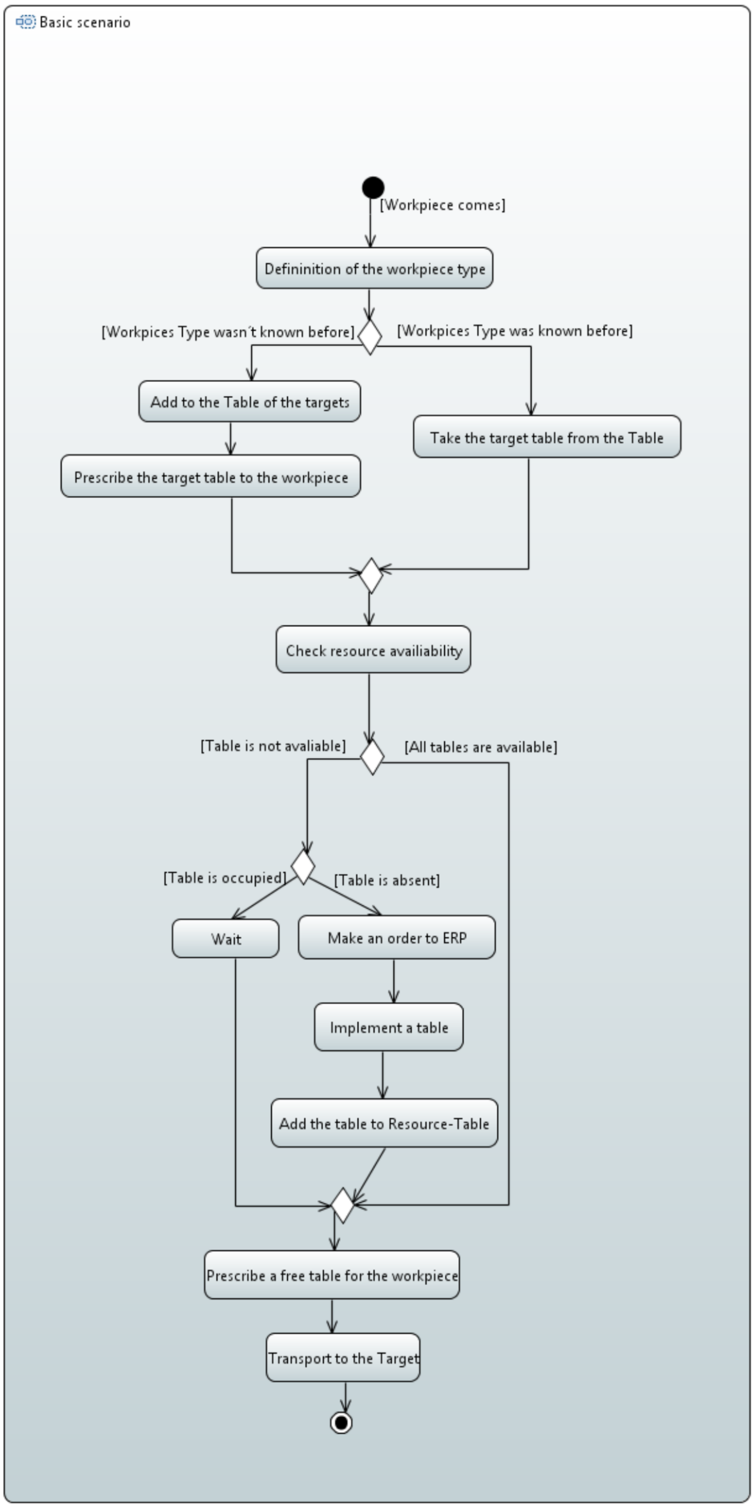


Figure 4.4Trigger diagramof operational scenarios (one cycle)

Now, all functions of the control architecture should be separated between agents into the structure showed on Fig.4.2. Thus, peripheral control system, data base, and main control system should exist. Remembering all the requirements that were named in the thesis the following table is organized (Table 4.2).

Table 4.2 Separation of particular tasks between agents

Category	Function	Agent
Basic properties	Exchanges data with sensors and actuators (peripheral control system)	Executive Agent
	Reschedules the paths for new types	Rescheduler Agent
	Dispatches the system and provides the rules to the executive agent	Dispatcher Agent
	Provides safety maintenance and fault tolerance: Back-up controller (“start”, “continue”, “stop”, “alarm”)	High-availability Agent
	HMI “start”, “continue”, “stop”, “alarm”	Supervisor Agent
	“Stop” / “alarm” by power interruption, voltage control	High-availability Agent
	Secure maintenance: leakage protection	High-availability Agent
	Inspection / examination of the system work	High-availability Agent Executive Agent
	Performing complex tasks such as: - Optimization of technological process parameters (belt speed or acceleration) - evaluation, classification and decision making in a process, interaction with a process model in the direction adaptation and adjustment of control rules	Supervisor Agent
	-order-related communication with ERP/MES (e.g. in the directions bottom-up – “order to ERP system, like buy a new table” - for the conveyor case, top-down -“order to do a customized detail” – for the robot case)	Supervisor Agent
	Element identification	Executive Agent
	Process execution knowledge	Executive Agent High-availability Agent Supervisor Agent
	Production knowledge	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Insurance of the availability of conveyor parts	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Provide the path of the element, routing	Dispatcher agent Rescheduler agent
	Cooperation with operator	Rescheduler agent Supervisor Agent
	Order plan	Dispatcher agent Executive Agent Supervisor Agent
	Resource visualization	Supervisor Agent
	Central control system	Supervisor Agent
	Knowledge base	Dispatcher Agent

Reached Flexibility	Process flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Product flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Routing flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Volume flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
Industrie 4.0 relation	Product customization	Dispatcher Agent Rescheduler Agent Supervisor Agent
CPS relation	Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time	Executive Agent Supervisor Agent
	Self-organization and adaptation due to resource accessibility (part of convoy are available)	Dispatcher Agent Rescheduler Agent Supervisor Agent

Short work description

The Executive agent (EA) performs simple control on the field control level and carries the information within the control system.

The Dispatcher agent (DA) plans and restructures the tables of resources and elements. To do that, it saves and eventually and constantly renews the information.

The Rescheduler agent (RA) provides new resource implementation in the system and together with SA plans the new control model of the process, according to types of the products, indeed the workpieces in the case, by appearance of a new workpiece type that did not exist in the table of elements, RA and HAA provide safety sustainable control.

In the case of unavailability of any resource of conveyor, also in Table of functions, the SA interacts with ERP and sends an order to buy a new resource. After the resource is available in the conveyor, the RA integrates it and adds the resource to the Table of elements with its prescribed functions, so that the DA can plan a new path of the workpiece, considering the new resource. Thereby, the control system keeps a record of available resources and meets the requirements of the Industrie 4.0. To describe the control system the UML modeling is used.

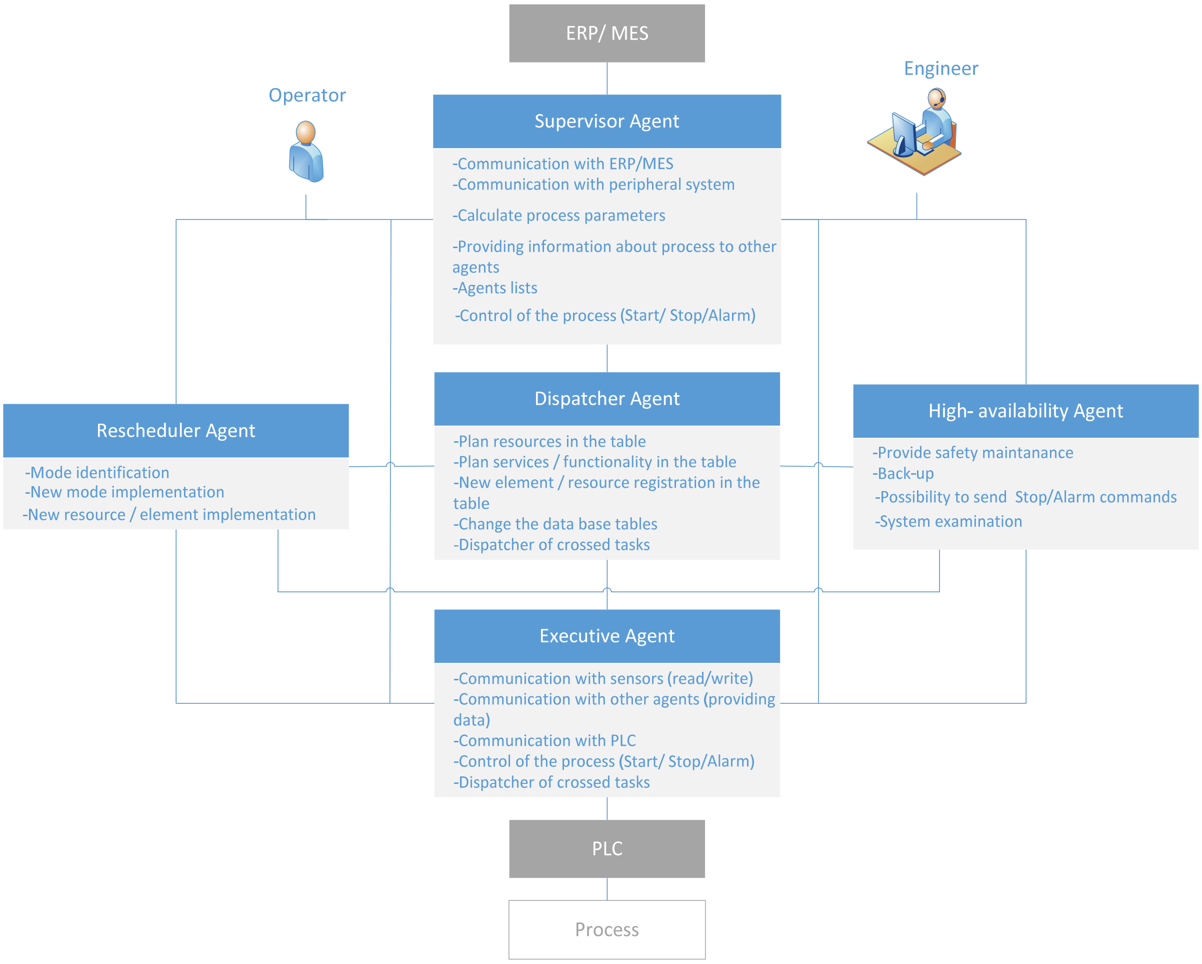


Figure 4.5 Controller system architecture by means of agents

In the end, summing up all information about agents´ description based on the architecture on Fig. 4.5 the following architecture is obtained. It was based on architecture given in Fig.4.2, where the control system represents the cooperation between supervisor and dispatcher agent. The peripheral control system from Fig.4.1 is influencedby the Executive agent. The dispatcher agent consists of Knowledge base, to which the Resheduler agent and operator have the access. External disturbances are mostly controlled by theHigh-availability agent. The entity of the data processing is presented by executive agent together with rescheduler agent. Both operator and engineer have the possibility to change the system.

The modeling of this application case is presented in more detail due to the fact that this application case of transportation system is chosen for the implementation, which will be held in the next Chapter.

For the system structure and behavior modeling of the control system, two languages will be used: Unified Modeling Language (UML) and System Modeling Language (SysML). UML is a graphical notation for the creation of object-oriented models for the analysis and the design of object-oriented software [Bal01]. SysML is the system modeling language created for System Engineering, a general-purpose graphical modeling language that supports and improves specifications, analysis, design, validation and verification of complex systems. It provides several diagrams that allow the description of different aspects of the model: requirements, behavior and structure of the system [Ant11, Wei06, Cal16]. SysML is defined as an extension of a subset of the UML. Both languages will be used for a better description of the system work in different information presentation.

During the modeling of a system different phases are usually considered. The diagrams have been implemented using Papyrus [PAP16], an Open Source UML tool:

- Analysis of the context, defining the system missions and purposes
- Definition of the system components and the interaction among them
- Description of the system functional behavior required to achieve these missions
- Allocation of the functions of each component

Using UML, essential aspects of control engineering issues such as concurrency, hardware/software architecture, and requirements traceability can be applied to multi-agent systems [Cal16¸ Alk11].

The selected diagrams in UML compose the high-level model of the Multi-Agent System are use case diagram, block diagram, and activity diagram for the different presentation of agent interconnections.

Here, a Use Case diagram has been chosen to describe the mission of the system. This diagram depicts system functionalities in terms of how the system is used by external actors to achieve its purposes. Figure 5.4 represents the context in which the system operates by defining how the system interacts with the external environment, i.e. the functions provided and the *actors* and other systems involved. It is used as the main starting point to define the structure and the functions of the multi-agent system using structural and behavioral diagrams.As actors ERP and MES, PLC and human help - operator and engineer interpositionare chosen (Fig.4.6).

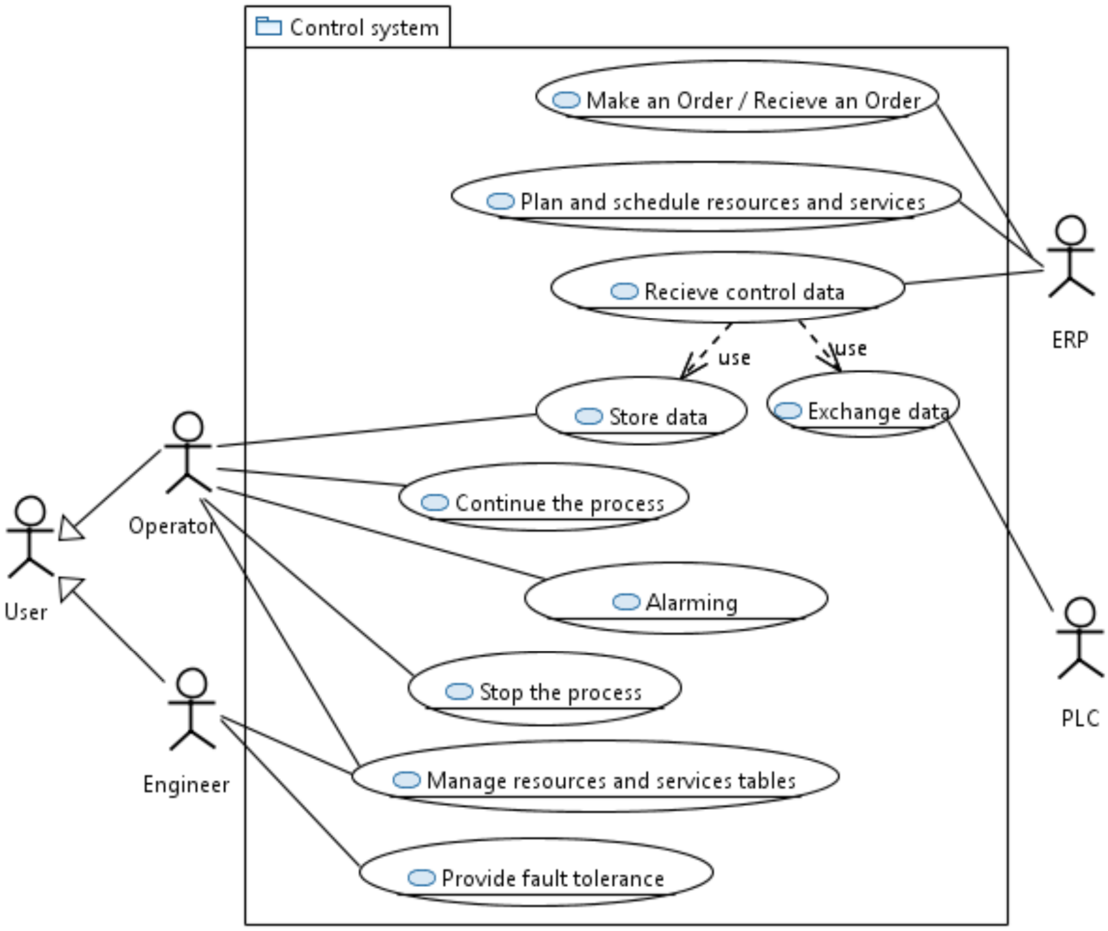


Figure 4.6 MAS Use Case Diagram

An Internal Block Diagram (IBD) is one of the structural diagrams provided by UML. It is mainly used to define the internal structure of a system in terms of how its internals parts are inter-connected, using ports and connectors, and what flows between them. It represents the components that compose the system as*blocks*and, through the *interfaces*, the information they exchange between each other.

Figure 4.7defines the internal structure of the multi-agent system and defines the general interaction between agents and direction of communication. The agents here represented are the ones identified within the DACS analysis. During the modeling phase, the flow of information within the multi-agent system is defined, as well as the inputs and outputs generated among the different agents and between the MAS and the external systems (actors).

Finally, behavioral diagrams are used to represent the functional modeling of the system. An Activity Diagram (AD) depicts the functional dynamic behavior of a system, by using activities or operations with input/output and control flow, analyzing step by step the actions performed. Partition lines group any set of activity or operation nodes based on their corresponding organizational units. The activity itself specifies the transformation of inputs to outputs through a controlled sequence of actions.

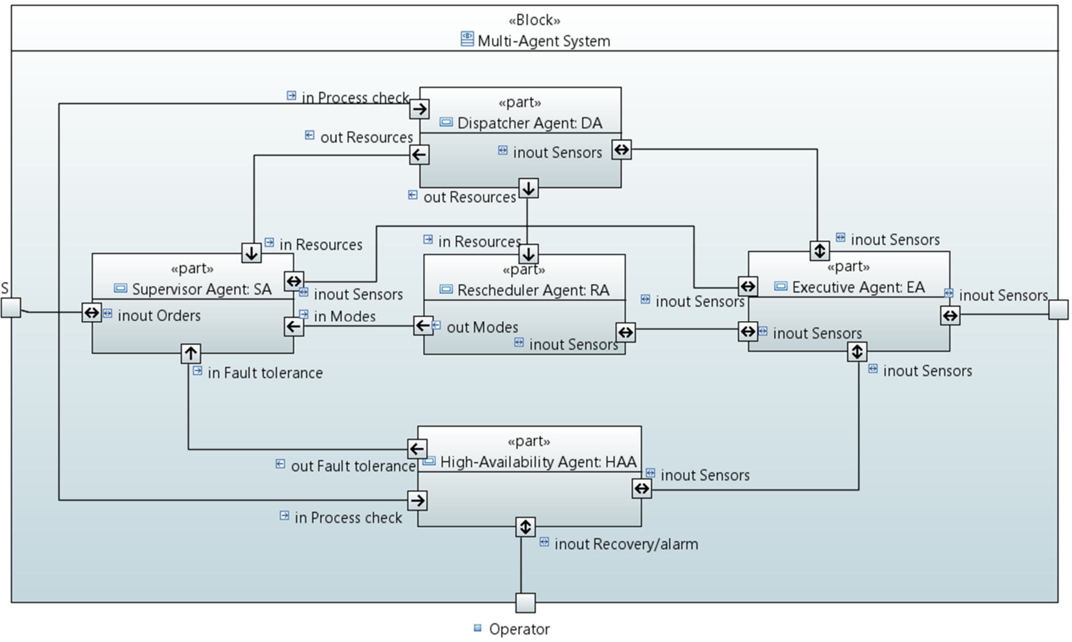


Figure 4.7 MAS Internal Block Diagram

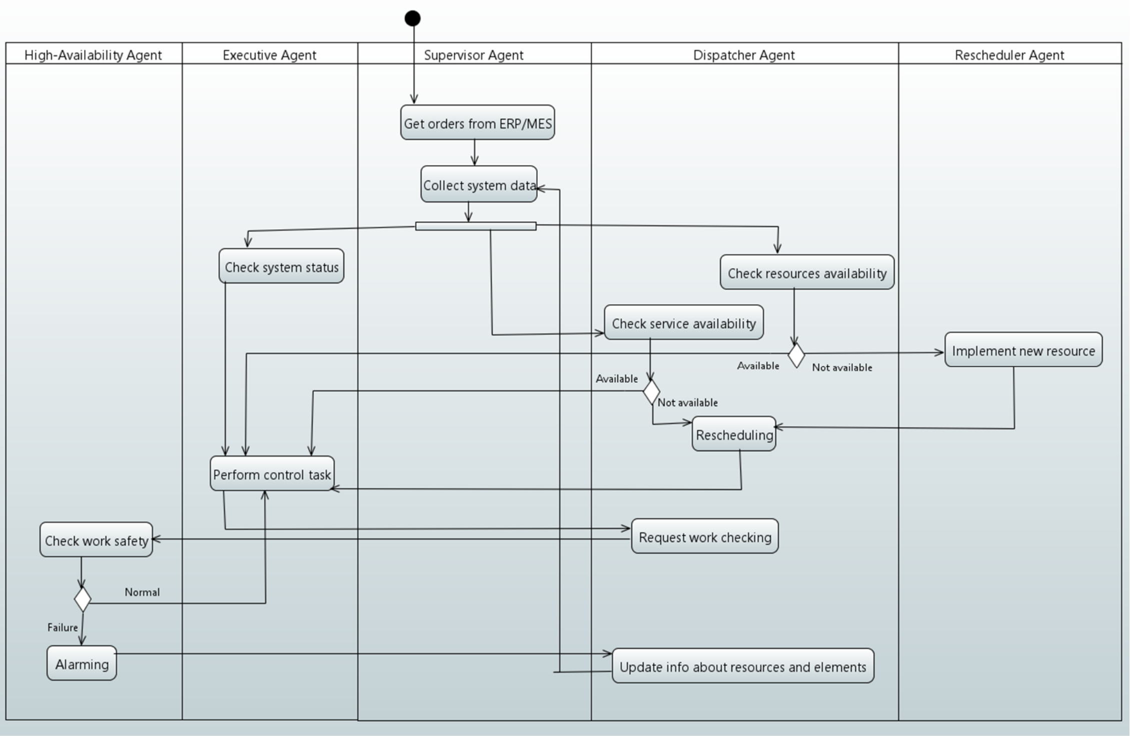


Figure 4.8MAS Activity Diagram

As depicted in a simplified version on Fig.4.8, the activity diagram of the MAS is used to model agents' interactions, describing the flow of control and flow of inputs and outputs among actions. These actions (functions) derive from the requirement analysis and the use case analysis, as they are necessary to achieve the goals of the system. This diagram also depicts “who” will execute that activity, using partitions lines, which address actions for each agent of the system.

It can be remarked that functional analysis of systems is completed defining State Machine Diagrams, which investigates all the states of the system operations and how the functions allow the transition between states, and also Sequence Diagrams, which describe the sequence of messages exchanged between actors and the system components as a function of time.

The modeling language is useful to completely define the system components and their features. This diagram representation describes the functional architecture and behavior of the multi-agent system taking into account the goals of the system and its requirements. Researchers [Ant11] relied on the Model Based System Engineering approach as it ensures a good quality of the system development process and enables the re-use of functions and components of the system. Moreover, thanks to the functional and physical analysis the number of functions and components included in the system will be kept low, only those really necessary are selected, reducing system complexity and the process development time.

As this application case was chosen for the final implementation in the thesis, the implementation description of the agents for this case using JADE will be provided in the next chapter.

4.3 “Heater” application case description with DACS methodology

According to the DACS methodology, the modeling of the steam production in the heater will be performed, agents will be defined.

At first, the physical process is defined as steam production through the combustion of a fuel.

Heater has a complicated architecture presented on Fig.4.9 and 4.10. As the input product, the pulverized lignite for combustion is considered. Through a dosing feeder lignite comes into a boiler. For the process control the data from a light sensor, which is placed before the burner, is used on the whole spectrum of a light (Fig.4.9). During the combustion,appear additional emissions of ash and slag, which are deposited partly in the boiler and during prolonged operation form an insulating layer between burner and boiler. For this reason, frequent boiler cleanings are necessary. The cleanings cause downtimes and additional costs that is why the optimization of the combustion process is needed.

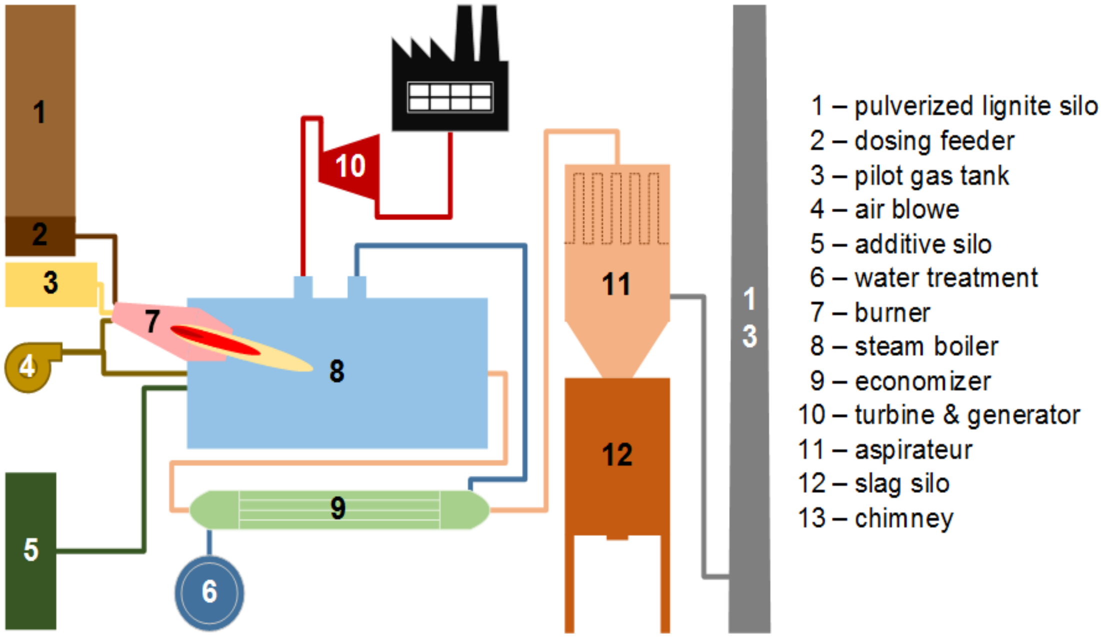


Figure 4.9Heater constituent elements

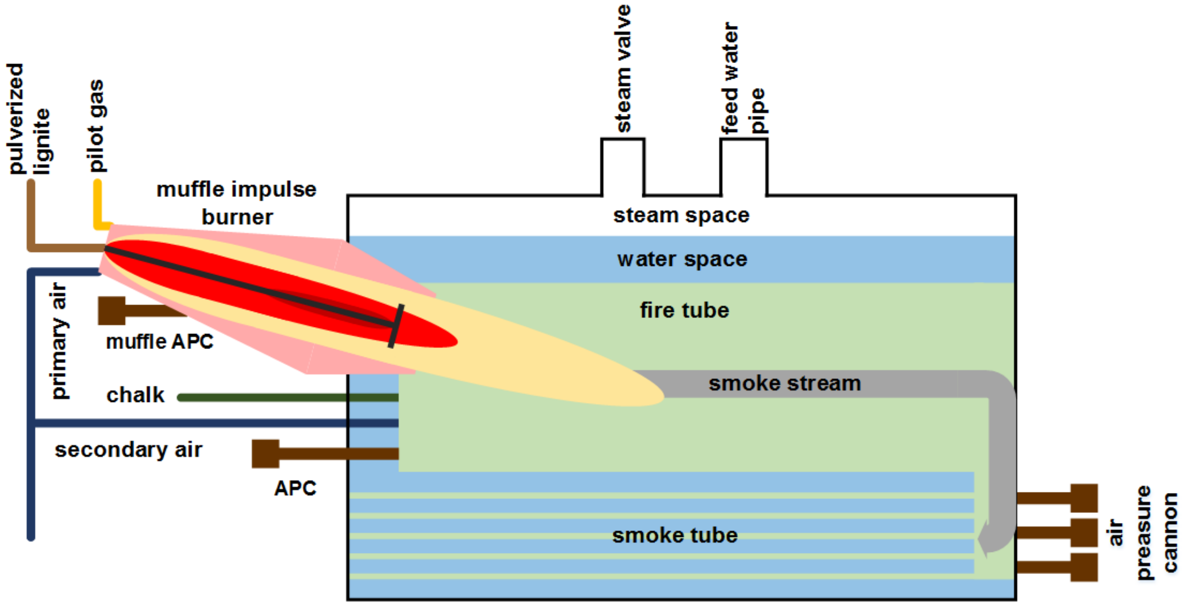


Figure 4.10Heater inside

a) The **production process** is defined as follows: the fuel, air and gas are supplied to a boiler. This process is controlled by a PID controller (Fig.4.11), which setting should be changed if the fuel is changed. If the settings are not correctly selected, the pollution of the tube occurs more often. Fuel provider changing happens from considerations of fuel costs or power saving factor, but, in the end, firms must pay more for the cleaning service.

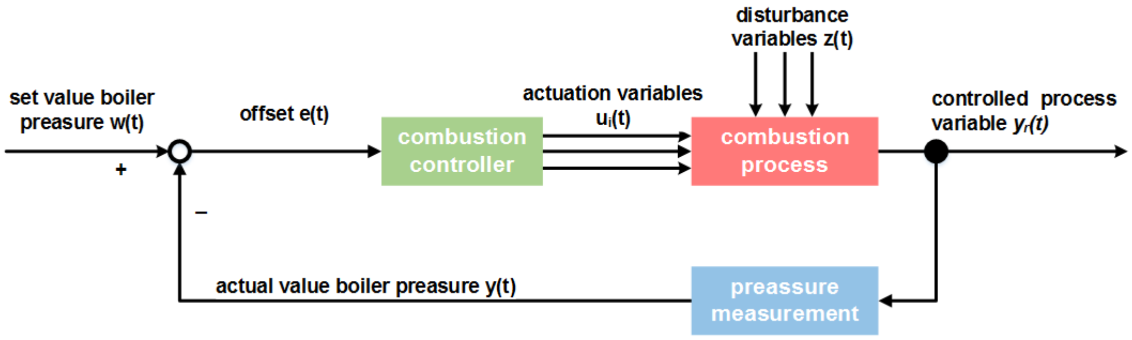


Figure 4.11A general control model of combustion process

b) The **operational conditions** specify the input and output of the production process. Input is a sensor data about fuel, air, gas amount, combustion process. Outputs are control values processed from a controller. **Disturbances** may occur any time during the processing, such as unavailability of the desired type of fuel or increase of the load of the boiler.

c) The production **goal**: efficiently and without interruption obtain the desired quality and quantity of heat/ energy.

d) **Requirements** to production are flexibility with respect to fuel changes, robustness with respect to failures, quality assurance, maintainability, existence of fallback strategies. These tasks will be considered further in the task table for different agents.

Again in this case, a solution to the production control problem is a control system, which controls a production and creates specified amount of energy, and which optimizes the performance of the boiler with respect to goals and requirements.

Thus, the objective is to model the heater control system, with opportunities to provide the same level of work, ensuring flexibility by fuel changes. For this purpose, an appropriate control rule should be chosen. At the same time, availability of heater parts is ensured, collecting and registering them in the control system.

Analysis of control decisions

Due to methodology, the control process is described. First of all, before boiler load all resources must be checked: valves, air filters, fuel system itself, piping system, feed water system, water softener. If resources are broken, or additional sensors should be implemented in the system, the resources should be ordered by ERP. After work start, the proper boiler temperature and pressure should be supervised according to boiler modes. Heater modes are divided into different modes, such as starting mode, stopping mode, energy-efficient mode, etc.. Modes could be changed due to a load of the boiler or modifiedfuel type. During the processing the adjusting of combustion air should be kept in relation to fuel flow. By the fuel changing controller parameters should be optimized to avoid the pollution in the boiler. The production scenarios are described in Table 4.3.

Table 4.3 Scenarios of the heater control

Use case	Scenario content
Pre-conditions	All resources and supplies are available
Main scenario	1. Check if all systems are in operating condition 2. If it is, start with first mode – base load 3. Define the mode according to fuel type 4. Select controller settings according to the mode, if the mode is known 5. Optimize control process due to the mode 6. Operation and sensor measurement 7. Unloading
Other scenarios	4.1. If the mode is unknown, count process model and control mode
Post-conditions	Customer receives desired product
Error-conditions	2.1. If it is not in operative conditions, repair system till the working condition 2.2. Order new resources to ERP system 2.3. Implement new resources

Dependencies between control decisions

- Operation is impossible without available resources or supplies.
- Operation is impossible without predefined modes and control parameters.

For the better process control supervision in the control system should receive as most as possible control data from sensors and actuators (Fig.4.12) [Rya15].

The steam production during the boiler functioning is a very complicated process. The simple ignorance can destroy the whole system and even can lead to victims. That is why the artificial intelligence, e.g. expert knowledge, is very important to use during the control. For that reason the control rules can be described according to each situation (each mode). If the situations / modes of heater are described with fuzzy logic, then as was advised in Chapter about AI, the fuzzy rules can be adjusted using other AI representations, such as neural networks and genetic algorithms.

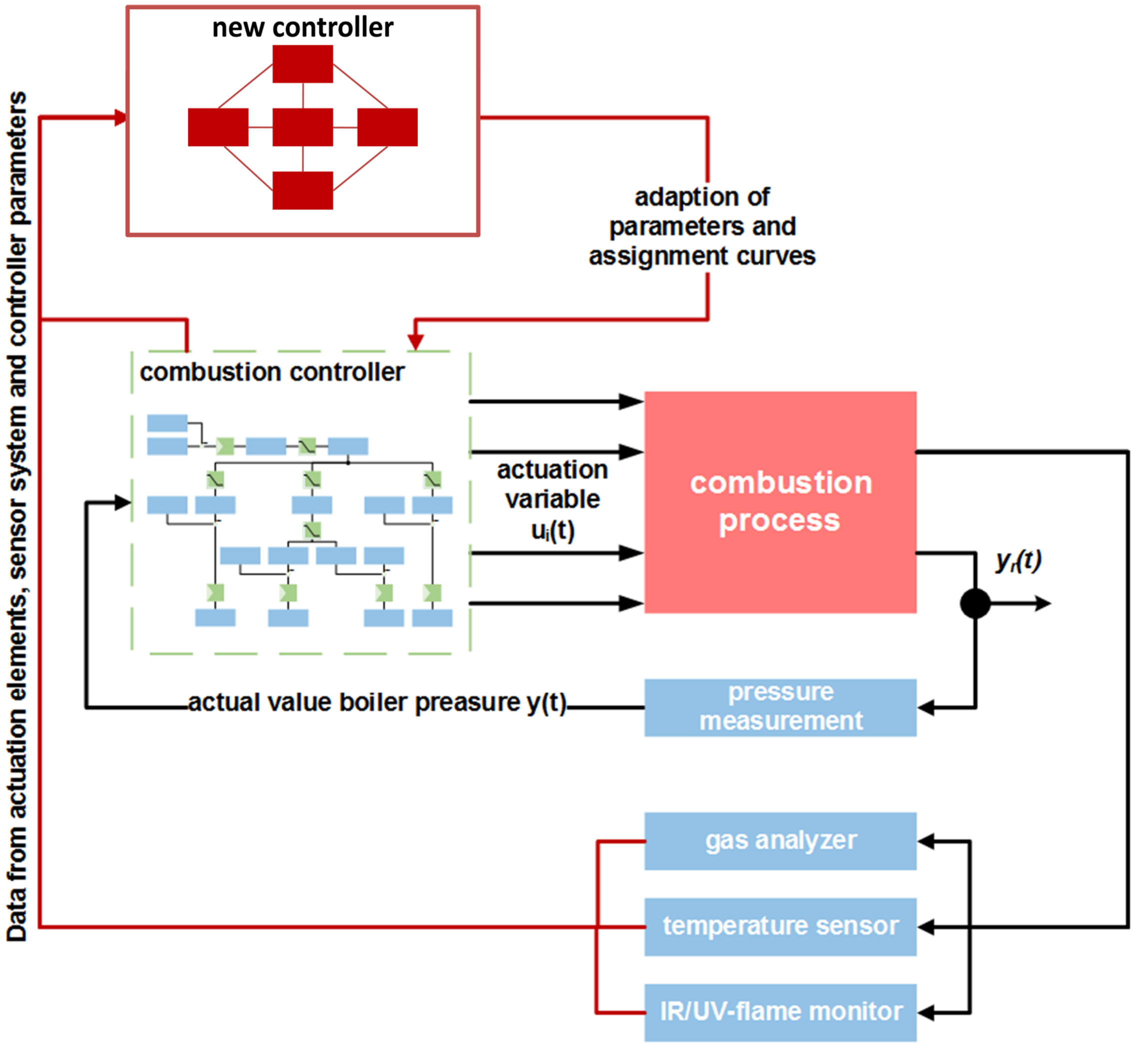


Figure 4.12 Additional controlthrough the MAS in the legacy control system

The additional AI will be used for the process optimization and placed above the legacy controller, and will promote self-adaptable system, which is able to function a long time without human help. The graphical representation is shown on Fig.4.12.

According to scenarios the trigger diagram is presented in Fig. 4.13. During the operation the mode of the boiler is defined cyclic. Thus, the production process is always optimized.

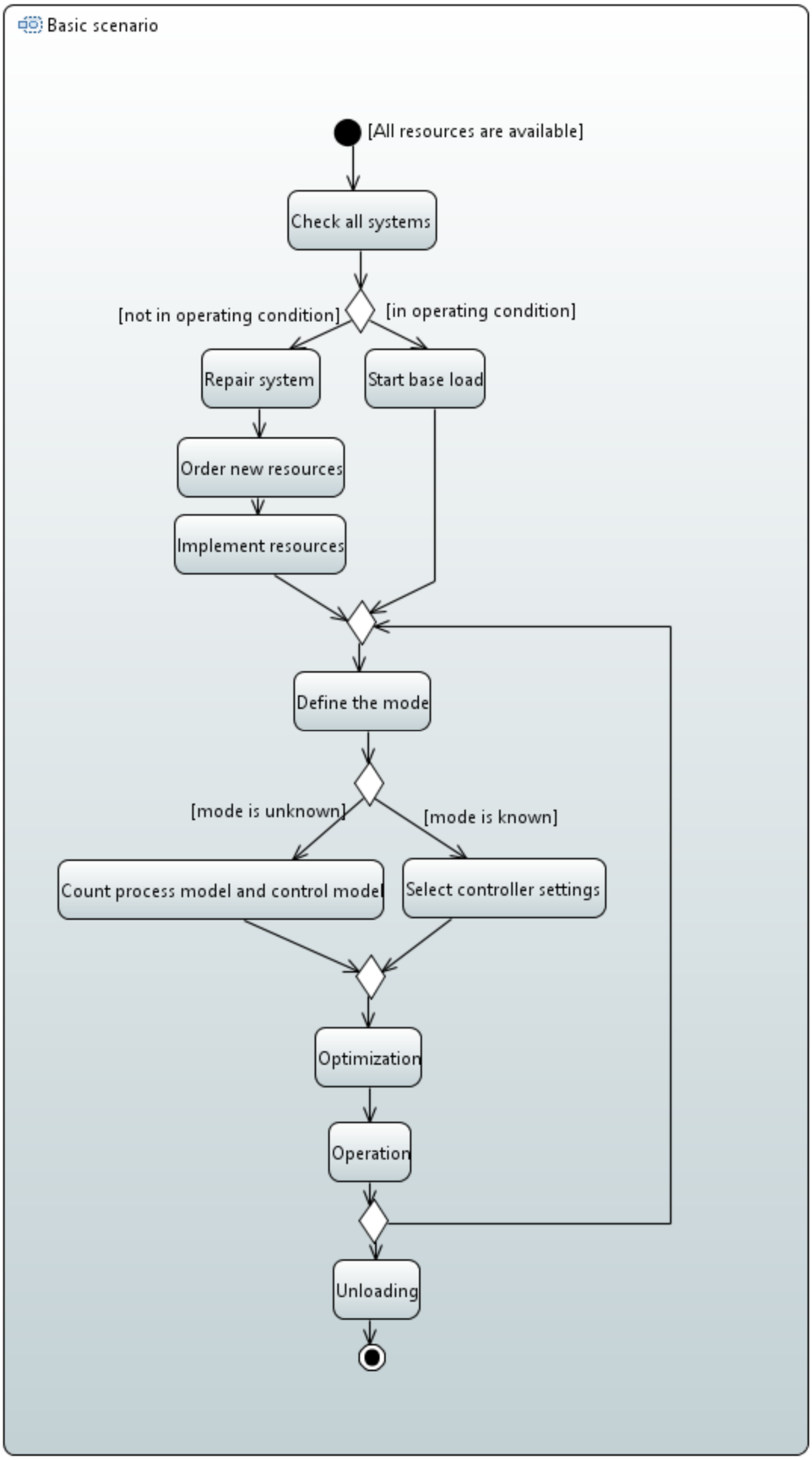


Figure 4.13Trigger diagram of operational scenarios

Identification of agents

Not to make it too complex for one agent, the control decisions should be suited to different agents, so that still the control problem will be solved, the modifications that improve the decision model are allowed. Some agents should be responsible for resources, other for model calculation and optimization. To perform the heat and steam production, the control system needs to know all information about available sensors.

Now, all functions of the control architecture should be separated between agents into the structure showed on Fig.4.2. Thus, peripheral control system, data base, and main control system should exist. Remembering all the requirements that were named in the thesis the following table is organized (Table 4.4).

Table 4.4 Separating particular tasks between agents

Category	Function	Agent
Basic properties	Exchanges data with sensors and actuators (peripheral control system)	Executive Agent
	Reschedules new modes and implements new resources	Rescheduler Agent
	Dispatches the system and provides the control rules to the executive agent	Dispatcher Agent
	Provides safety maintenance and fault tolerance: Back-up controller (“start”, “continue”, “stop”, “alarm”)	High-availability Agent
	HMI “start”, “continue”, “stop”, “alarm”	Supervisor Agent
	“Stop” / “alarm” by power interruption, voltage control	High-availability Agent
	Secure maintenance: leakage protection	High-availability Agent
	Inspection / examination of the system work	High-availability Agent Executive Agent
	Performing complex tasks such as: - Optimization of technological process parameters (PID controller) - evaluation, classification and decision making in a process, interaction with a process model in the direction adaptation and adjustment of control rules	Supervisor Agent Dispatcher Agent
	Communication with ERP system	Supervisor Agent
	Mode identification according to sensor signals	Executive Agent
	Process execution knowledge	Executive Agent High-availability Agent Supervisor Agent
	Production knowledge	Dispatcher Agent Rescheduler Agent Supervisor Agent

	Insurance of the availability of resources	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Provide the controller settings for the mode	Dispatcher agent Rescheduler agent
	Cooperation with operator	Rescheduler agent Supervisor Agent
	Order plan, plan of orders	Dispatcher agent Executive Agent Supervisor Agent
	Resource visualization	Supervisor Agent
	Central control system	Supervisor Agent
	Knowledge base	Dispatcher Agent
Reached Flexibility	Processflexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Product quality flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Steam quantity flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
Industrie 4.0 relation	Product customization	Dispatcher Agent Rescheduler Agent Supervisor Agent
CPS relation	Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time	Executive Agent Supervisor Agent
	Self-optimization due to resources and fuel supply	Dispatcher Agent Rescheduler Agent Supervisor Agent High-availability agent

Short work description

The Executive agent (EA) performs simple control on the field control level and carries the information within the control system.

The Dispatcher agent (DA) plans and restructures the tables of resources (if needed) and modes of the boiler. To do that, it saves and eventually and constantly renews the information.

The Rescheduler agent (RA) provides new resource implementation in the system and together with SA plans the new or adjust the control model of the process, according tochanging environment, by the appearance of a new mode type that did not exist in the Table of modes.

The main task of the Supervisor Agent (SA) is mathematical calculations for process model and model adjustment. In the case of unavailability of any resource of boiler, e.g.

broken sensor, the SA interacts with ERP and sends an order to buy a new resource. After the resource is available in the boiler, the RA integrates it and adds the resource to the Table of elements with its prescribed functions/services. Further the normal functioning of the boiler is expected with continuous self-optimization through Supervisor Agent.

The High-availability (HAA) is responsible for the stable work of the control system; it examines system and starts back-ups.

Thereby, the control system keeps a record of available resources and meets the requirements of the Industrie 4.0. To implement the control system the UML modeling is used.

In the end, summing up all information about agents´ description based on the architecture on Fig. 4.14 following architecture is obtained. It was based on the architecture described in Fig.4.2, where the control system represents the cooperation between supervisor and dispatcher agent.

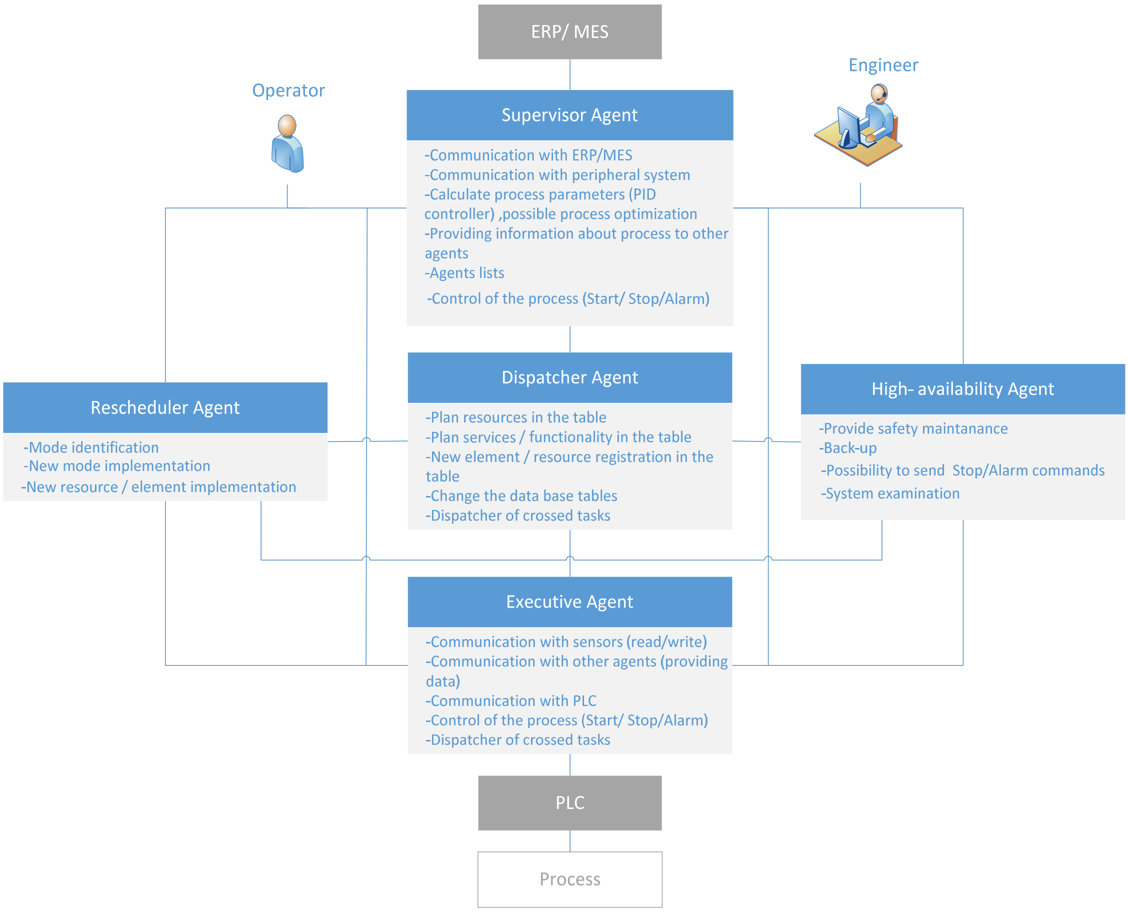


Figure 4.14The agent architecture for a boiler control system

The peripheral control system from Fig.4.1 is influenced by the Executive agent. The Dispatcher agent consists of Knowledge base, to which the Resheduler agent and operator have the access. External disturbances are mostly controlled by the High-availability agent. Entity of the data processing is presented by the Executive agent together with rescheduler agent. Both operator and engineer have the possibility to change the system. It is evident that also in this case the agent set is the same, executes the same main functions. But before this architecture is called the general unique control architecture another application case "assemble robot" should be considered.

The Use Case Diagram depicts control system functionalities in terms of how the system is used by external actors to achieve its purposes. Figure 4.15represents the context in which the system operates by defining how the system interacts with the external environment, i.e. the functions provided and the *actors* and other systems involved. As actors ERP and MES, PLC and human help - operator and engineer interpositionare chosen.

Through this diagram, the mission of the system that has to be modeled is well defined and described, and it is used as the main starting point to define the structure and the functions of the multi-agent system using structural and behavioral diagrams. Now it is evident that also cases in the Use Case Diagram are similar to the Use Case Diagram of a Conveyor case.

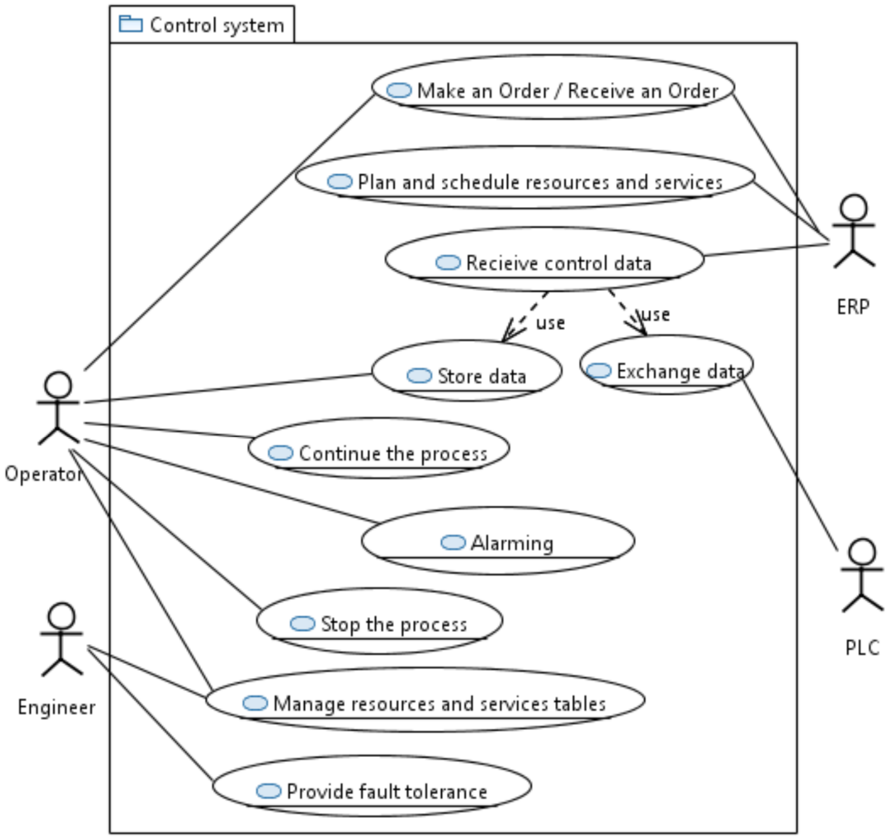


Figure 4.15 MAS Use Case Diagram

The Activity Diagram and Internal Block diagrams are absolutely the same way describe interactions between agents(Fig.4.7 and Fig.4.8). In next paragraph, the third application case will be described.

4.4 “Assemble robot” application case description with DACS methodology

According to the DACS methodology, at the beginning the physical production process should be described. The production process is the production of the desired detail (or a car as on Fig. 4.16) in the robotic cell, pursuing the flexibility, depending on the desired output and set of parts, and proving its availability, and choosing the right control settings (for example, drilling depth and diameter or point to place glue dots).

For the car production, in general, several operations should be carried out such as drilling or gluing. These operations could be performed by several robots (like KUKA robot on Fig. 4.17) at the station in a shop floor that are able to change their heads. To do that, all heads and their positions should be known.

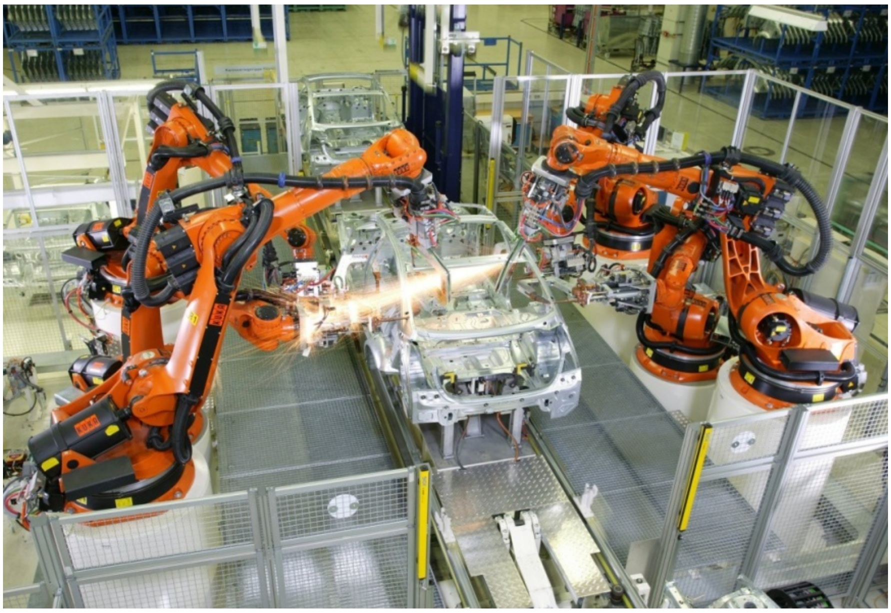


Figure 4.16Car production process in robotic cell according to [KUK16]

Figure 4.17 KUKA robot according to [AUC16]

Specification of the production control problem

a) The **production process** is defined as follows: the order specified by a customer comes from EPR system. The materials and resources to execute the order should be provided. In the case of a car production, it can be an order for a special carcolor or car body type. In the case of detail production it is a sawn or drilleddetail.

b) The **operational conditions** specify the input and output of the production process. Inputs are available resources and materials with an order. Output is a desired detail or a desired product. **Disturbances** may occur at any time during the processing, such as unavailability or occupation of the resources to perform the task.

c) The production **goal**: efficiently and without interruption obtain the desired product.

d) Production **requirements**: flexibility with respect to resource changes (e.g. changed supplies) and robustness with respect to failures, quality assurance, maintainability, existence of fallback strategies. These tasks will be considered further in the task table for different agents.

Again, in this case, a solution to the production control problem is a control system, which controls a production and creates specified products, and which optimizes the performance of the assembly robot with respect to goals and requirements.

Thus, the objective is to model the assembly robot control system, with opportunities to provide desired sophisticationin manufacturing of detail. For this purpose, an appropriate control rule should be prescribed, e.g. choosing diameters and drilling speeds of the drill of a robot. At the same time availability of resources should be ensured, collecting and registering them in the control system.Further, the production scenarios will be described in Table 4.5.

Table 4.5 Scenarios of the assemble robot producing a desired product

Use case	Scenario content
Pre-conditions	All resources and supplies are available
Main scenario	1. Get an order 2. Plan the resources 3. Prepare to start a work, if all resources in the job shop 4. Plan the supplies (materials/water/oil) 5. Prepare to start, if the supplies are already in the job shop 6. Start the operation 7. Collect data from sensors 8. Process optimization
Other scenarios	2.1. If the resource is not available, order it to ERP 2.2. Implement the resource 4.1. If the supply material is not available, order it to ERP 4.2. Implement it
Post-conditions	Customer receives desired product
Error-conditions	2.1. If the resource is availablebut not in the job shop, find his location in the Table of resources/services 2.2. Implement it 4.1. If the supply material is availablebut not in the job shop, find his location in the Table of resources/services 4.2. Implement it

Analysis of control decisions

To produce the desired product assembly robot should be aware of all supplies, materials and resources that are needed to start a production process. For that reason in the system data base with all resources and services of the system exist. This data is provided according to the ideas of Industrie 4.0, saying that “Industrie 4.0-component” is a component with communication capability and it has at least “individually-known” grade (Chapter 2.3.3) .

Dependencies between control decisions

- Operation is impossible without available resources.
- Operation is impossible without available supplies.

The trigger diagram of agent scenarios is presented in Fig. 4.18. During the operation the self-optimization of the production process takes place.

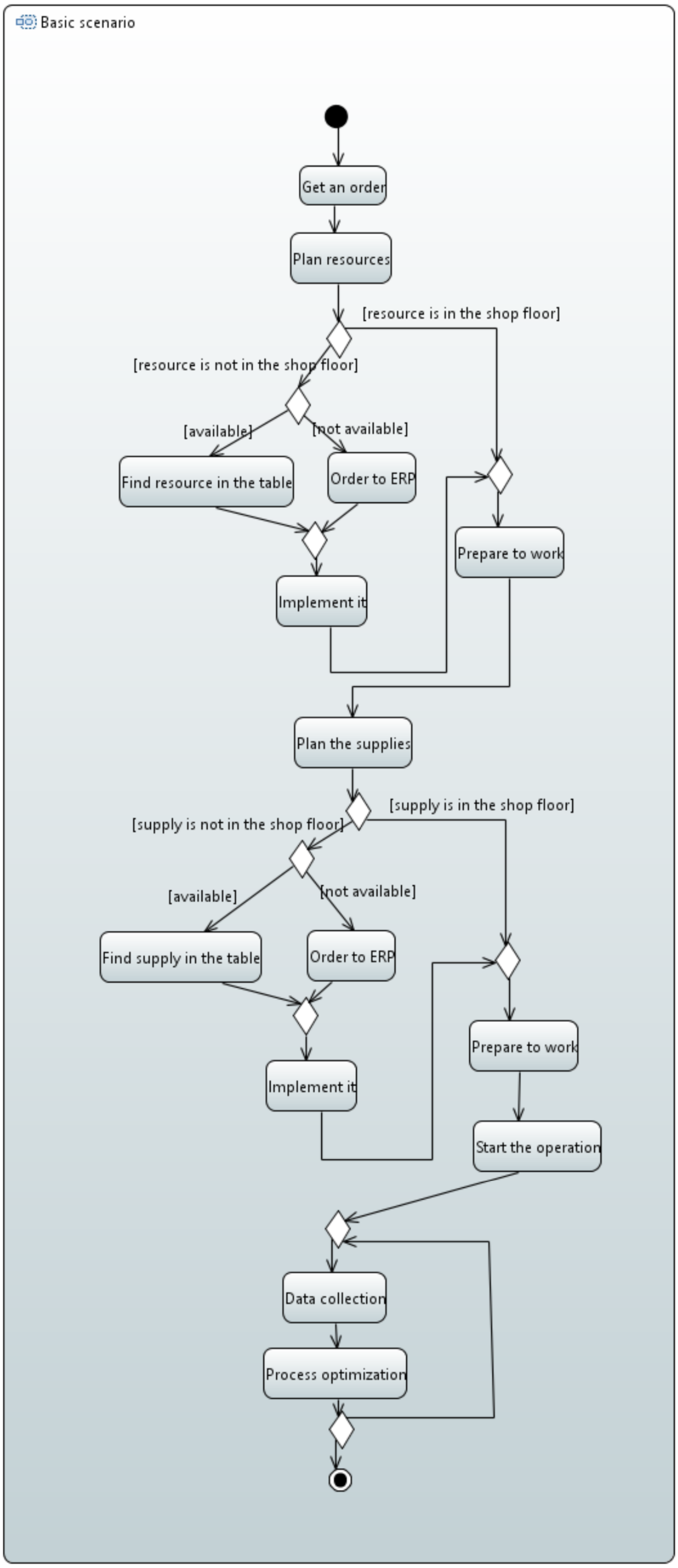


Figure 4.18 Triggerdiagram of operational scenarios

Identification of agents

Not to make it too complex for one agent, the control decisions should be suited to different agents, so that still the control problem will be solved, the modifications that improve the decision model are allowed. Some agents should be responsible for resources, other for model calculation and optimization. To perform the production, the control system needs to know all information about available sensors and resources.

Now, all functions of the control architecture should be separated between agents into the structure showed on Fig.4.2. Thus, peripheral control system, data base, and main control system should exist. Remembering all the requirements that were named in the thesis the following table is organized (Table 4.6).

Table 4.6 Separating particular tasks between agents

Category	Function	Agent
Basic properties	Exchanges data with sensors and actuators (peripheral control system)	Executive Agent
	Reschedules and implements new resources	Rescheduler Agent
	Dispatches the system and provides the control rules to the executive agent	Dispatcher Agent
	Provides safety maintenance and fault tolerance: Back-up controller (“start”, “continue”, “stop”, “alarm”)	High-availability Agent
	HMI “start”, “continue”, “stop”, “alarm”	Supervisor Agent
	“Stop” / “alarm” by power interruption, voltage control	High-availability Agent
	Secure maintenance: leakage protection	High-availability Agent
	Inspection / examination of the system work	High-availability Agent Executive Agent
	Performing complex tasks such as: - Optimization of technological process parameters (PID controller, e.g. speed of a drill) - Evaluation, classification and decision making in a process, statistical tests results, work with process model or control model and sharing data with DA	Supervisor Agent Dispatcher Agent
	Interconnection to other enterprise layers such as: Order-related process steps (down- top order to ERP system, like “need a drill” - for the robot, top-down -“order a desired product” from the robot). Orders types: materials, quantity, cost resource utilization (service description), workers.	Supervisor Agent
	Mode identification according to sensor signals	Executive Agent
	Process execution knowledge	Executive Agent High-availability Agent Supervisor Agent
	Production knowledge	Dispatcher Agent Rescheduler Agent Supervisor Agent

	Insurance of the availability of resources	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Provide the controller settings for the chosen work type (drilling speed for special material)	Dispatcher agent Rescheduler agent
	Cooperation with operator	Rescheduler agent Supervisor Agent
	Order plan, plan of orders	Dispatcher agent Executive Agent Supervisor Agent
	Resource visualization	Supervisor Agent
	Central control system	Supervisor Agent
	Knowledge base	Dispatcher Agent
Reached Flexibility	Processflexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Product quality flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
	Product quantity flexibility	Dispatcher Agent Rescheduler Agent Supervisor Agent
Industrie 4.0 relation	Product customization	Dispatcher Agent Rescheduler Agent Supervisor Agent
CPS relation	Parallel data collection, data fusion, processing of physical data from the environment, locally, globally and in real-time	Executive Agent Supervisor Agent
	Self-optimization due to resources and materials supply	Dispatcher Agent Rescheduler Agent Supervisor Agent High-availability agent

Short work description

The Executive agent (EA) performs simple control on the field control level and carries the information within the control system.

The Dispatcher agent (DA) plans and restructures the tables of resources for the detail production. To do that, it saves and eventually and constantly renews the information.

The Rescheduler agent (RA) provides new resource implementation in the system and together with SA plans the new control model of the process, according to types of the products, indeed the detail in the case, by appearance of a new resourcethat did not exist in the Table of available parts of a robot, RA and HAA provide safety production and add it to the Table of resources.

The Supervisor Agent (SA) is responsible for the mathematical calculations such as process model and control model calculations. In the case of unavailability of any resource of the robot, the SA interacts with ERP and sends an order to buy a new resource (e.g. drill, oil). After the resource is available in the robot, the RA integrates it and adds the resource to the Table of elements with its prescribed functions/services. Further, the normal functioning of the robot is expected with continuous self-optimization.

The High-availability (HAA) is responsible for the stable work of the control system; it examines system, starts back-ups.

Thereby, the control system keeps a record of available resources and meets the requirements of the Industrie 4.0. To implement the control system the UML modeling is used.

In the end, summing up all information about agents´ description based on the architecture on Fig. 4.19 following architecture is obtained. It was based on the Fig.4.2, where the control system represents the cooperation between supervisor and dispatcher agent.

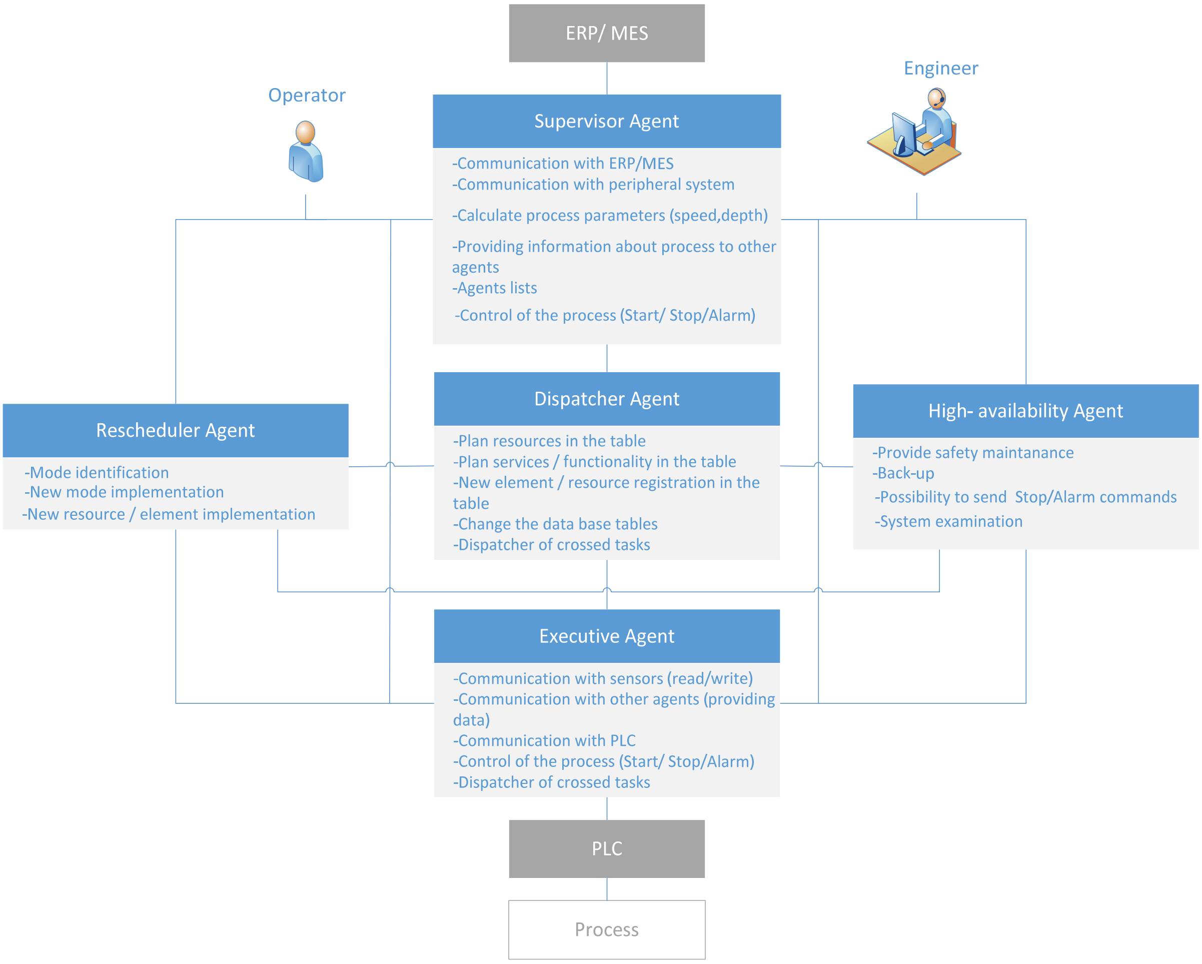


Figure 4.19The agent architecture for assembly robot control system

The peripheral control system from Fig.4.1 is influenced by the Executive agent. The Dispatcher agent consists of Knowledge base , to which the Resheduler agent and operator have the access. External disturbances are mostly controlled by the High-availability agent. The entity of the data processing is presented by the Executive agent together with the Rescheduler agent. Both operator and engineer have the possibility to change the system. It is evident that also in this case the agents set is the same, executes the same main functions as in two cases before.

The Use Case Diagram depicts control system functionalities regarding how the system is used by external actors to achieve its purposes. Figure 4.19represents the context in which the system operates by defining how the system interacts with the external environment. As actors ERP and MES, PLC and human help - operator and engineer interpositionare chosen (Fig.4.20).

Through this diagram, the mission of the system that has to be modeled is well defined and described, and it is used as the main starting point to define the structure and the functions of the multi-agent system using structural and behavioral diagrams. Now it is evident that also cases in the Use Case Diagram are similar to the Use Case Diagram of a Conveyor case (Fig. 4.7 and Fig. 4.8).

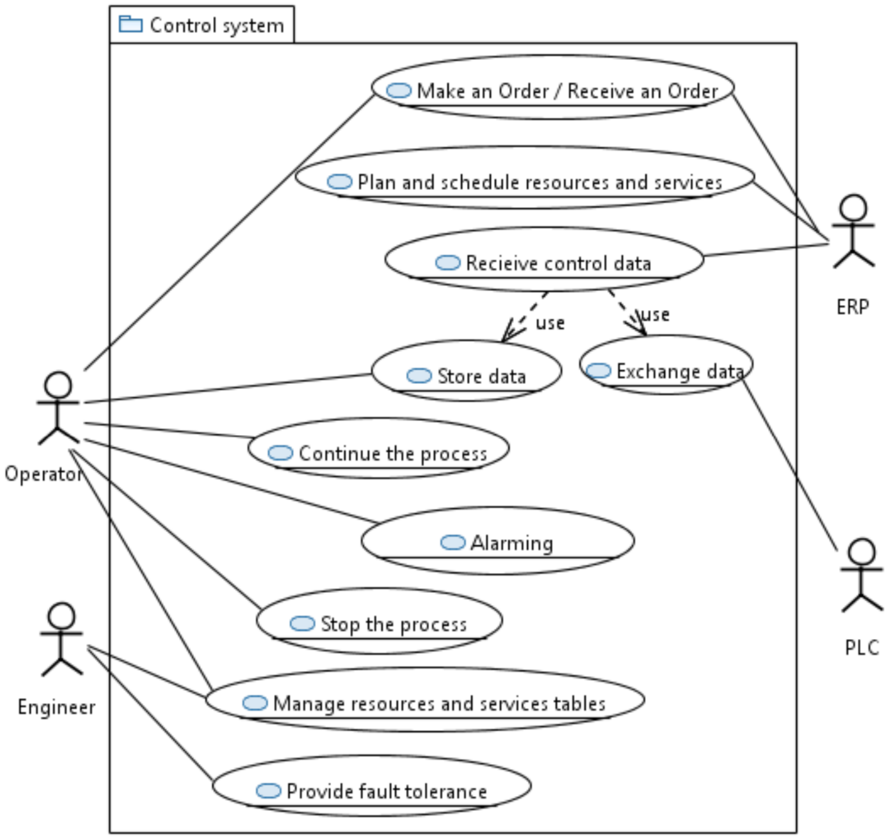


Figure 4.20 MAS Use Case Diagram

Summing up, all three cases got the same control architecture comprising five agents same in each case. That means that it is possible to call the architecture as the universal architecture for production processes control that includes five agents with general tasks, which will be applicable to other production jobs with further adjustment of parameters. And so, the 4th question of the thesis is answered. Provided analyze gives the right to claim the architecture as a universal architecture, which includes all necessary function (according to DACS analyze), which can be adjusted to further production jobs.

In the next Chapter, the next steps of the control system implementation will be provided.

5.Phase of technical product development.Phase of product implementation

The goal of Chapter 5 is to describe the implementation of MAS to the small Fischertechnik production plant. Each agent will be described and implemented with behaviors in JADE,behaviors will be presented as Class diagram and as Activity diagram.

5.1 Mechatronic-/electronic planning. Assembly

As it was considered, the implementation of the agent-based control architecture for chosen application case “Conveyor” will be shown in this Chapter. The implementation of MAS is based on Fischertechnik-Based Laboratory unit, which is depicted in Fig.5.1, upgrading existing controllers by extension to Industry 4.0 Components ideas. Implementing the control system it is worth to say that 80% of its cost relate to existing equipment. In the system, there is already a PLC Raspberry Pi with IEC-61131 runtime environment to control the plant (Fig.5.1), a bus couplerfrom Phoenix Contact, a switch, PC and power supply. According to the settings, PLC could be used for the back-up opportunities and at the same time all agents could be realized on the PC. Hardware structure is presented in Fig. 5.2.

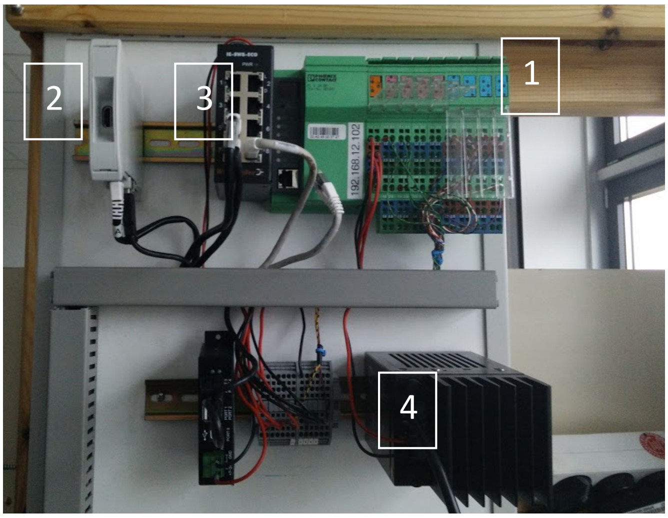


Figure 5.1 Control architecture: 1 – Bus coupler, 2- PLC Raspberry Pi presenting a legacy system, 3- Switch, 4- Power Supply

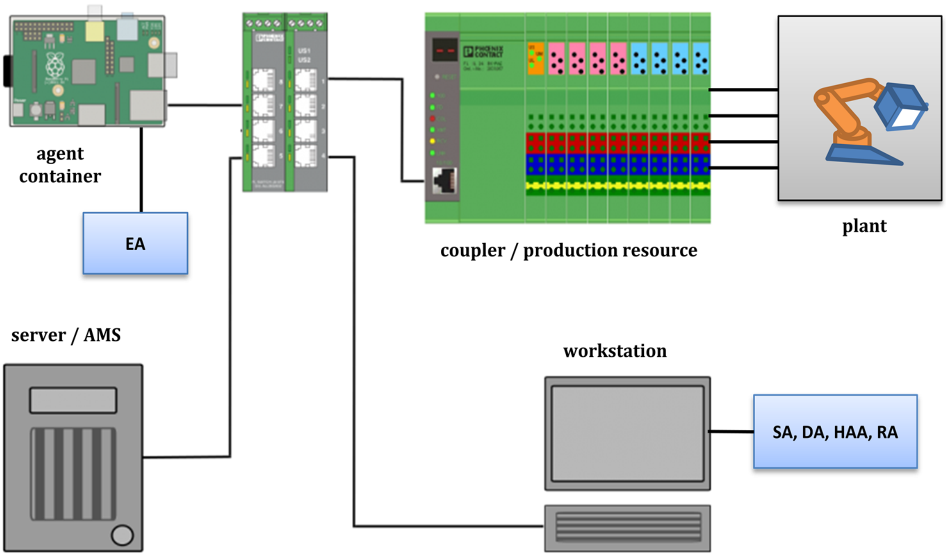
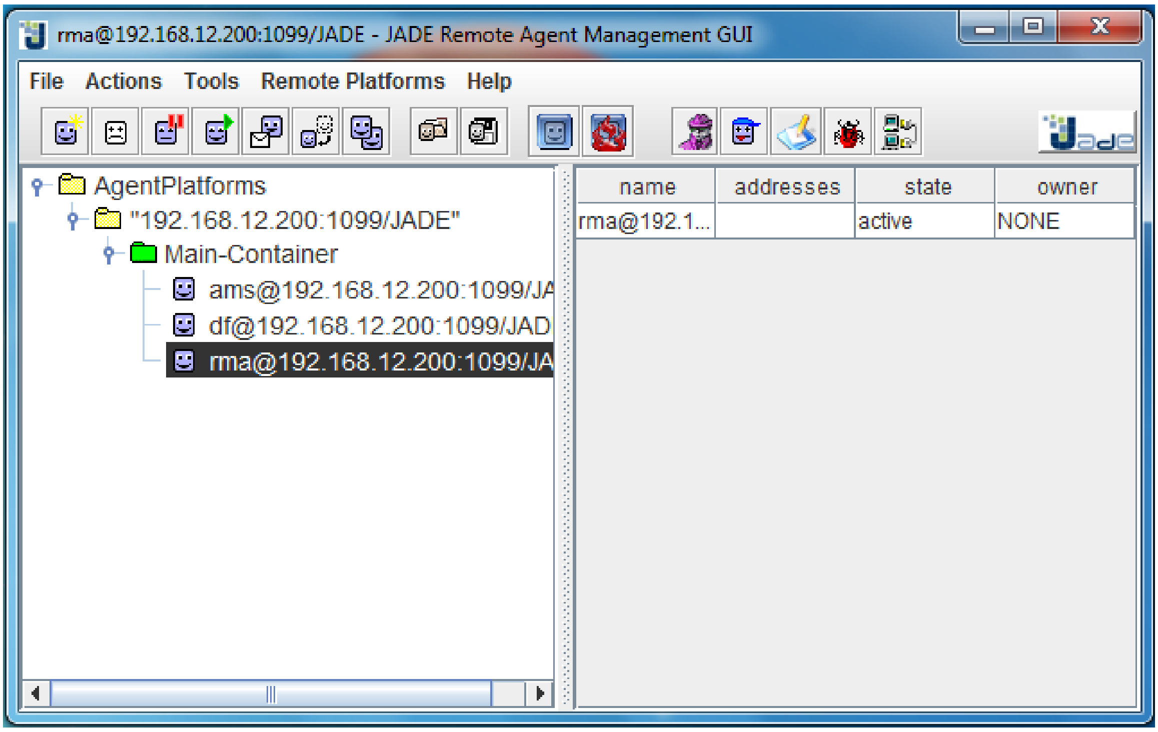


Figure 5.2Hardware structure including agents

The control architecture consists of a PLC with a legacy control system, a switch in the middle which connects all of the elements together, a bus coupler, a PC and server- agent container. The container provides agents communication for all agents registered there. Realization of Container in Jade is presented on Fig.5.3.

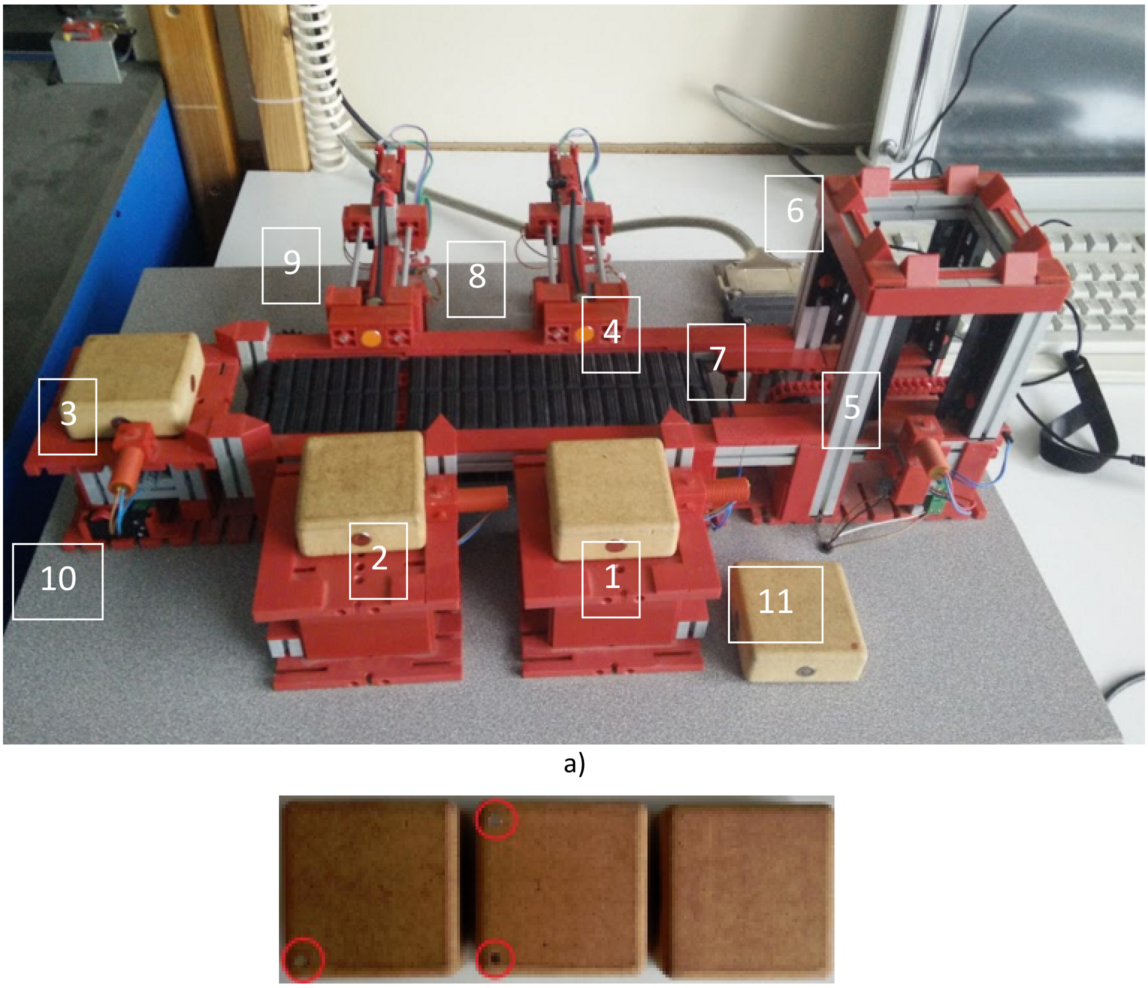


AgentPlatforms
  "192.168.12.200:1099/JADE"
    Main-Container
      ams@192.168.12.200:1099/JA
      df@192.168.12.200:1099/JAD
      rma@192.168.12.200:1099/JA

name	addresses	state	owner
rma@192.1...		active	NONE

Figure 5.3 Agent container AMS

In more details Conveyor with its elementsis presented on a Fig.5.4(a). Also on the Fig. 5.4(b) the wooden test blocks with magnets are shown. The meaning of conveyor sensors and function of resourcesis defined in Table 5.1.



a)

b)

Figure 5.4 Transportation system from Fischertechnik(a) and blocks with magnets (b)

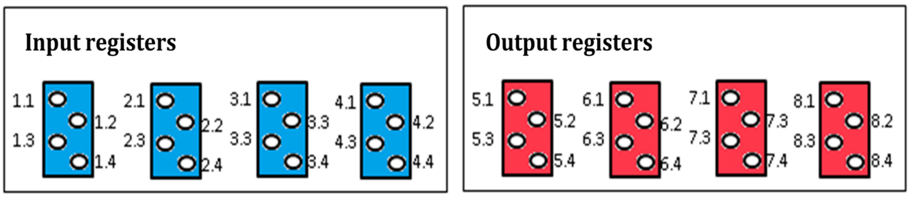


Figure 5.5Pins in Bus coupler

Thesignals generated by sensors are transferred from the manufacturing plant to PC through Bus coupler. Agents and Bus coupler communicate in both directions via Modbus TCP/IP providing a self-optimized control system.The programming code of the control architecture is realized in Java by means of Eclipse Software. Agents are distributed between a normal PC and PLC, and registered in a JADE Container with their IP address. On the Raspberry Pi PLC the Linux Operating System runs and on PC –a usual Windows 7 OS with Eclipse.

Table 5.1 Components of the plant and their states

Nr	Object	Sensors	Function
1	**Table 1**	1 sensor	occupied/ not occupied
2	**Table 2**	1 sensor	occupied/ not occupied
3	**Table 3**	1 sensor	occupied/ not occupied
4	**Transporting Band**	Motor	Move till object is on band/ Stop
5	**Small band**	Motor 1 sensor, 1 magnet	Move till magnet meets sensor/ Stop Cycle done
6	**Loading station**	2 sensors	occupied/ not occupied
7	**Object definition place**	2 sensors for magnets	Definition of the object
8	**Pusher 1**	2 sensors Motor	Position 1, Position 2 Move to Position 2
9	**Pusher 2**	2 sensors Motor	Position 1, Position 2 Move to Position 2
10	**Button**	On/Off	No function
11	**Work piece**	4 marks 1 or 2 magnets	Position on a table Object definition

Table 5.2 Themeaning of pins in Bus coupler module

PIN	Meaning
1.1	Button (№10 in Table 5.1)
1.2	inductive sensor from small band
1.3	inductive sensor for block identification (left side)
1.4	inductive sensor for block identification (right side)
2.1	Position of Pusher 1 (out)
2.2	Position of Pusher 1 (in)
2.3	Position of Pusher 2 (out)
2.4	Position of Pusher 2 (in)
3.1	Inductive sensor on Pusher 1
3.2	Inductive sensor on Pusher 2
3.3	--
3.4	--
4.1	Light cell
4.2	Inductive sensor for Table 1
4.3	Inductive sensor for Table 2
4.4	Inductive sensor for Table 3
5.1	Motor for small band
5.2	Motor for big band
5.3	--
5.4	--
6.1	Motor for Pusher 1 (out)
6.2	Motor for Pusher 1 (in)
6.3	Motor for Pusher 2 (out)
6.4	Motor for Pusher 2 (in)
…	Other unassigned pin

As Bus coupler from Phoenix Contact [PHO16] for the Modbus/TCPprotocol has 16 digital outputs and 16 digital inputs, pinsmeaning is shown on Fig. 5.5 and in Table 5.2.

The bus coupler registers are divided following way: blue are input register Reg0 and red is an output resister Reg.384, as shown on the Fig.5.6. The sensor values are sent in a string way presentation as “SensData” variable.

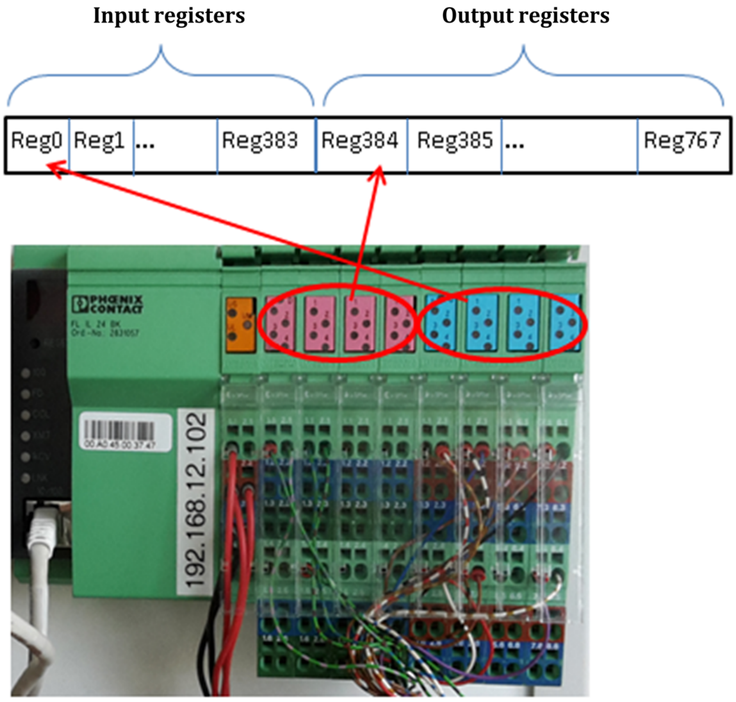
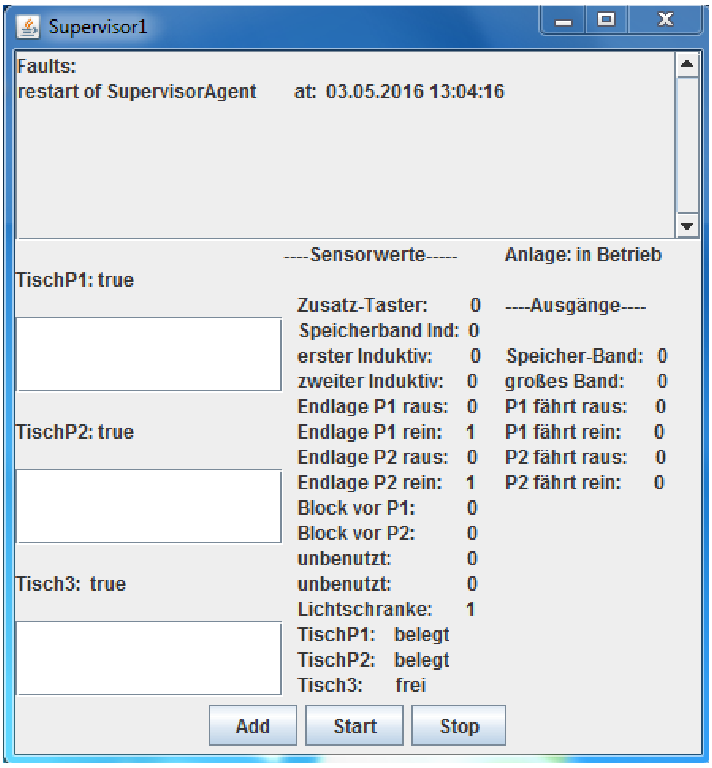


Figure 5.6 Bus couplerdepiction with Inputs (blue) and Outputs(red) and appropriate registers

The transportation system is directly controlled by a low-cost PLC runtime based on Raspberry Pi computers and Buscoupler from Phoenix Contact, as well as a normal PC that enables the communication exchange with the operator through agraphical user interface (GUI) (Fig.5.7, Fig.5.8). The GUI is described in next paragraph.

5.2 HMI development

The control system has more than one human interface. Fig. 5.7represents the GUI for the operator. Using this small GUI, the operator can add available tables in the list of tables (on the example there 3 tables) and can send “STOP”, “START” signals to the system. Also, the operator receives the information about each sensor of the Fischertechnik model (from Table 5.2)andoutput bytes of motors, so as alarming information about “back-uped” agents.

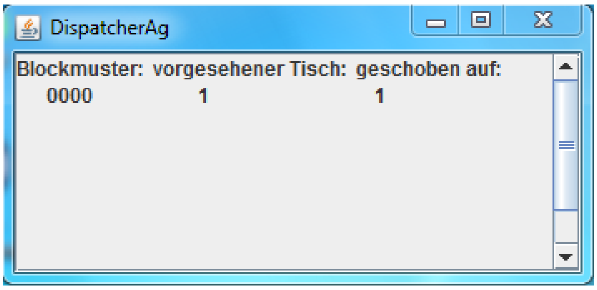


```
Faults:
restart of SupervisorAgent      at:  03.05.2016 13:04:16

                         ----Sensorwerte-----      Anlage: in Betrieb
TischP1: true
                           Zusatz-Taster:      0    ----Ausgänge----
                           Speicherband Ind:  0
                           erster Induktiv:     0    Speicher-Band:  0
                           zweiter Induktiv:   0    großes Band:     0
                           Endlage P1 raus:   0    P1 fährt raus:    0
TischP2: true              Endlage P1 rein:    1    P1 fährt rein:     0
                           Endlage P2 raus:   0    P2 fährt raus:    0
                           Endlage P2 rein:    1    P2 fährt rein:     0
                           Block vor P1:        0
                           Block vor P2:        0
                           unbenutzt:            0
Tisch3:  true              unbenutzt:            0
                           Lichtschranke:      1
                           TischP1:   belegt
                           TischP2:   belegt
                           Tisch3:     frei
```

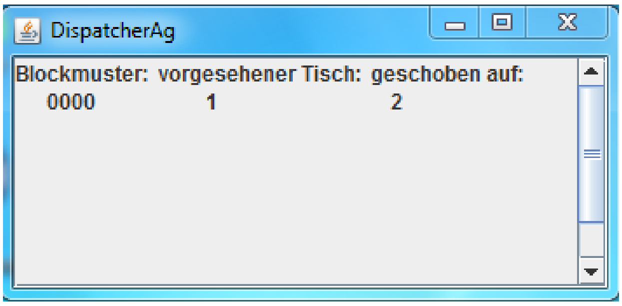
Figure 5.7 Main User Interface GUI

The operator will see any information about agents´ functioning. On the example (Fig.5.7), GUI shows that the Supervisor Agent was restarted (to show the work of GUI), also the date and time of this operation.



```
Blockmuster:  vorgesehener Tisch:  geschoben auf:
    0000              1                        1
```

a)



```
Blockmuster:  vorgesehener Tisch:  geschoben auf:
    0000              1                        2
```

b)

Figure5.8 Second GUI showing the type of block, prescribed destination and reached table

Another GUI presented on Fig.5.8 shows to operator the type of defined block (“000”), the destination table 1, and in the case of the table is free (a), he will arrive on prescribed table (the Table 1), if the table is occupied (b) he will arriveat the auxiliary table (Table 2). More information is in next paragraph.

5.3 Controller programming. Behaviors in Multi-Agent System

5.3.1 Supervisor Agent

Supervisor agent was implemented having an advantage in making important decisions. The given Fischertechnik model has no band speed regulation, so the optimization of the speed and PID parameters were not considered. But in future, with other models, it could be possible.

SA has a GUI to provideto the operator all important details about plant operation(Fig. 5.7).

The Supervisor Agent together with its graphical user interface (GUI)is the interface between people and the Jade agents. Via cyclic requests to EA values from PLC are transmitted and shown in the GUI. The three input boxes (Fig.5.7) tables are signed by "true" (table is not available) or "false" (table is available) enters. By pressing the "Add" button, the given tables transmitted to the DispatcherAgent,as the contents example: "101" is given, the first character: "1" = Table 1 is available, second character: "0" = Table 2 is not available, third character: "1" = Table 3 is available. By pressing the "Start" button the start command is sent to the EA (contents of the message: "1") and by pressing the "Stop" button, the stop command issent to the EA ("0"). Furthermore, should the High-Availability Agent (HA) fail unexpectedly, this is restarted by SA.

The main activities are connected with the GUI, operating with messages from- and to agents. Further, behaviors and activities of SA will be described using Class and Activity Diagrams.

The behaviors of the Agent are shown in the Table 5.3.

Table5.3SupervisorAgent´s main parameters and Behaviors

	Methods in a Class Diagram	**Represents Activity**
Behaviors in JADE	UpdateAgentList()	Update the available agents list
	setStart()	Setting the START command for Execute Agent, if the START button was pressed
	setStop()	Setting the STOP command for Execute Agent, if the STOP button was pressed
	setStartParams()	Define the existing tables by adding of "True" or "False" in the GUI, when ADD button was pressed
	writeMessage()	Prepare and send messages to the DA(template [XXX]) and EA (about START/STOP events)
	getSensorSignals()	Sensor data collection from EA
	CheckMailBox()	Check messages from high-availability agent
	restartHighAvailAgent()	Restart HAA
	startAgents()	Launch of agents
	updateSensorSignals()	Executes getSensorSignals()
Parameters (Properties in a Class Diagram)	Data:String	Pattern of 3 tables from GUI, template[XXX] operator gives
	StartStopData:String	Value of pressed button START and STOP [x]
	SensorsData:String	Values of all sensors, template [XXXXXXXX_XXXXXXXX_]

On the Fig.5.9 the presentation of the Supervisor Agent in the view of a Class Diagram in Papyrus Software is shown.

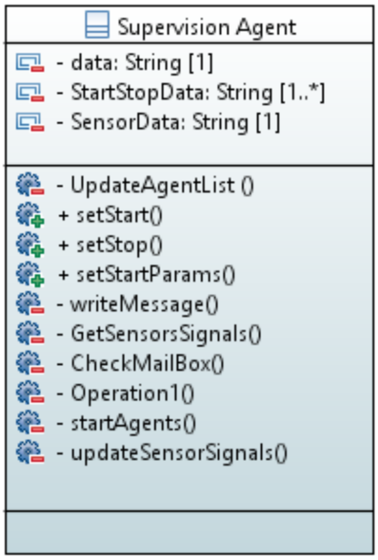


Figure 5.9The Class Diagram for Supervisor Agent

For better describing of each implemented behavior further for each behavior a table with sufficient information and ActivityDiagram is presented in the attachment.Activity diagram presenting the general work of the Supervisor Agent implemented in JADE is showed on Fig.5.10.

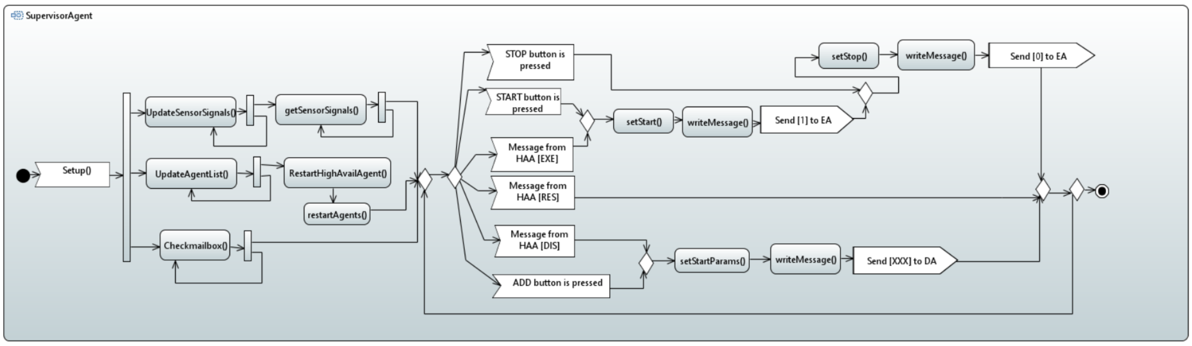


Figure 5.10 The general Activity Diagram for Supervisor Agent

5.3.2 Dispatcher Agent

The main job of the DA in the system is to find the destinationspaths to the blocks due to the defined block´s type. The block pattern obtained from EA is compared with the patterns that are already stored. If the pattern already exists, the stored with this pattern table is sent to the EA (content of this message: either "00" goal is Table1; "01" = target is Table 2; "10" = target is Table 3 or "11 "= no table free).

Table 5.4 Dispatcher Agent main parameters and Behaviors

	Methods in a Class Diagram	Represents Activity
Behaviors in JADE	UpdateAgentList()	Update the available agents list
	checkMailBox()	Processing incoming messages from other agents (EA, SA)
	FindBlockDestination()	Determine the target position of the current block
	WriteMessage()	Preparing and sending messages to EA, RA
	CheckMailBoxInTime()	Waiting function before CheckMailBox()
Parameters (Properties in a Class Diagram)	T_p1:boolean	Value for table 1
	T_p2:boolean	Value for table 2
	T_3:boolean	Value for table 3
	Typen:string	Types of blocks
	Ziele:string	Destination paths
	Ausweich:string	Substitution tables

If the pattern has not been stored by block, a new entry with this pattern is created. In another case, the determination of the target table is taken from the ReschedulerAgent. For that DA sends a message to the RA with the pattern from the block and the states of the three tables (for example, "1010 | 101"). After the target table was determined by the RA, the DA sends a message to the EA with the target table (for example: "10" = target is “Tisch3”).After the DA gets a message from the SA,if the GUI of SAof "Add" button has been pressed. This message consists of three characters (for example, "101"), which represent the input tables. They are required for the determination of the target table, because tables were initialized as "occupied" also cannot be selected as a target. The behaviors of the Agent are shown in Table 5.4.

On the Figure 5.11 the presentation of the Supervisor Agent in the view of a Class Diagram in Papyrus Software is shown.

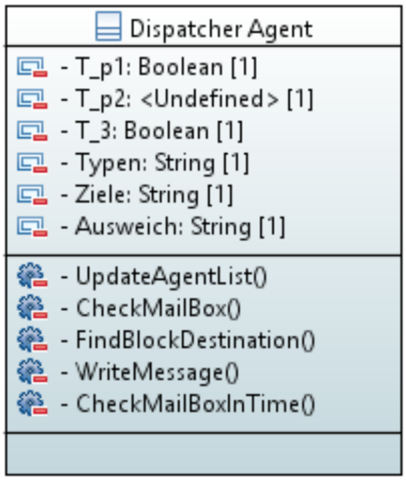


Figure 5.11The Class Diagram for Dispatcher Agent

For better describing of each implemented behavior further for each behavior a table with sufficient information and ActivityDiagram is presented in the attachment.

Activity diagram presenting the general work of the Dispatcher Agent implemented in JADE is showed on Fig.5.12.

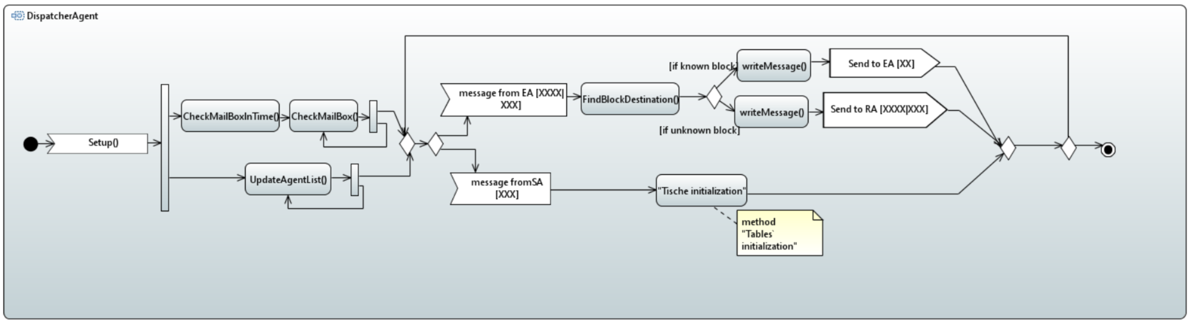


Figure 5.12 The Activity Diagram for Dispatcher Agent

5.3.3 Rescheduler Agent

As it was mentioned, the job of implemented Rescheduler Agent is to prescribe the new destination path for the new type of the defined magnets-template on the block and send the destination back to the DA, if it was requested.

The ReschedulerAgent accepts the allocation of the target table for the obtained from the DA block pattern. The RA receives from the DA, in the case of a new block pattern, a message with the block pattern and the states of the three tables (e.g ., "1010 101"). After the target table was determined by the RA, it sends a message with the block pattern and the target table back to the DA (e.g ., "1010 S 10"). The last two characters match the target table and the first four characters of the block pattern. Separately both strings are " S ". The behaviors of the Agent are shown in Table 5.5.

Table 5.5ReschedulerAgent main parameters and Behaviors

	Methods in a Class Diagram	**Represents Activity**
Behaviors in JADE	UpdateAgentList()	Update the available agents list
	checkMailBox()	Processing incoming messages from DA
Parameters (Properties in a Class Diagram)	*Private*RAAntwort:string[1]	Answer of Rescheduler Agent

On the Fig. 5.13 the presentation of the Rescheduler Agent in the view of a Class Diagram in Papyrus Software is shown.

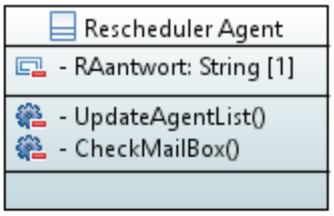


Figure 5.13The Class Diagram for Rescheduler Agent

For better describing of each implemented behavior, further for each behavior a table with sufficient information and ActivityDiagram is presented in the attachment.

Activity diagram presenting the general work of the Rescheduler Agent implemented in JADE is showed on Fig.5.14.

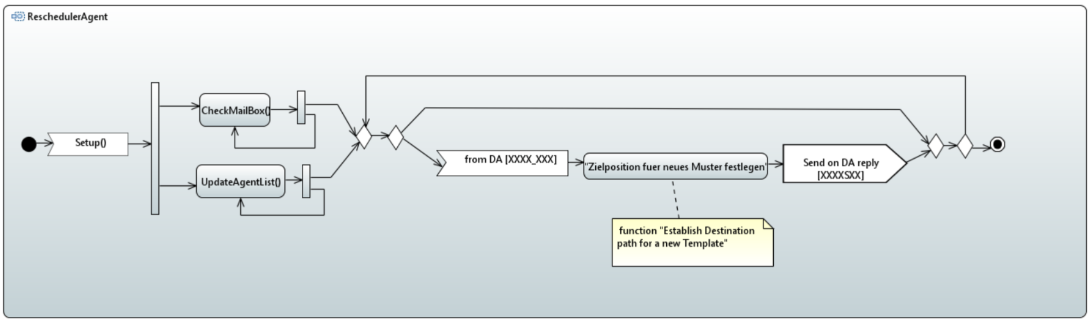


Figure 5.14The Activity Diagram for Rescheduler Agent

5.3.4 Executive Agent

Executive Agent represents the field control level and communication with bus coupler. The executive agent can be placed on the Pi controller representing the legacy system and keeping all the instructions for the blocks inside (destinations). Another option is to leave the legacy system on Pi controller and through the EA send instructions to the control system that controlled the transportation model before.

The Executive Agentis responsible for the conveyor system control. He sends requests to the PLC via a Modbus TCP / IP connection to view the currently active inputs (sensors) and outputs (motors) to write the outputs to activate the motors. To make available the process of readingthe values by PLC in real time, these requests are executed cyclically in the millisecond range. Then, EA provesthe obtained in a ticker behavior, whether a “start” or “stop”commands was receivedfrom the Supervisor Agent (SA). This message consists of a character (either "1" = “start” signal or "0" = “stop” signal). In case of a “start” signal, the "Control" -Behavioris started. Here blocks are automatically moved out of the storage and added the induced inductive sensors. The blocks are identified this way.

After identifying a message to the DispatcherAgent (DA) will be sent including the pattern of the block and the status of the three tables. This message is as follows: for example, "1010 | 110". The first four characters are the patterns from the block ("1010") and the last three characters are the states of the tables ("110" means: first character: "1" = Table 1 is occupied, second character: "1" = Table 2 is occupied, third character: "0" = Table 3 is not occupied). By a "|" character both strings are optically isolated from each other so that they are easier to identify. The behaviors of the Agent are shown in Table 5.6.

On the Figure 5.15 the presentation of the Executive Agent in the view of a Class Diagram in Papyrus Software is shown.

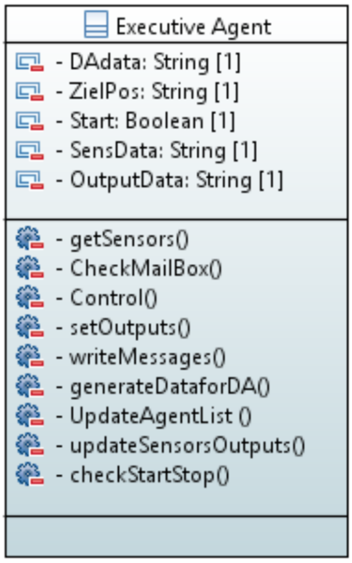


Figure 5.15The Class Diagram for Executive Agent

Table 5.6Executive Agent main parameters and Behaviors

	Methods in a Class Diagram	**Represents Activity**
Behaviors in JADE	UpdateAgentList()	Update the available agents list
	getSensors()	Read the sensors´ values from PLC
	getOutputs()	read the output registersfrom PLC
	checkMailBox()	Processing incoming messages from DA, SA
	Control()	Conveyor system is put into operation
	setOutputs(intval)	Write output registers of the PLC
	writeMessage()	Prepare and send messages to DA
	generateDataForDA()	Generating the content of the message to DA from the triggered sensors
	updateSensorsOutputs()	Update the sensor values and engine states
	checkStartStop()	Verification of the start / stop commands (obtained by SA)
Parameters (Properties in a Class Diagram)	*Private*DAdata:string	Data for DA [XXXX|XXX]
	*Private*ZielPos:string	Destination path from DA [XX]
	*Private*start:boolean	“Start” command from SA [X]
	*Private*SensData:string	Data from sensors [XXXXXXXX_XXXXXXXX_]
	*Private*OutputData:string	Data from output registers

For better describing of each implemented behavior, further for each behavior a table with sufficient information and ActivityDiagram is presented in the attachment.

Activity diagram presenting the general work of the Executive Agent implemented in JADE is showed in Fig. 5.16.

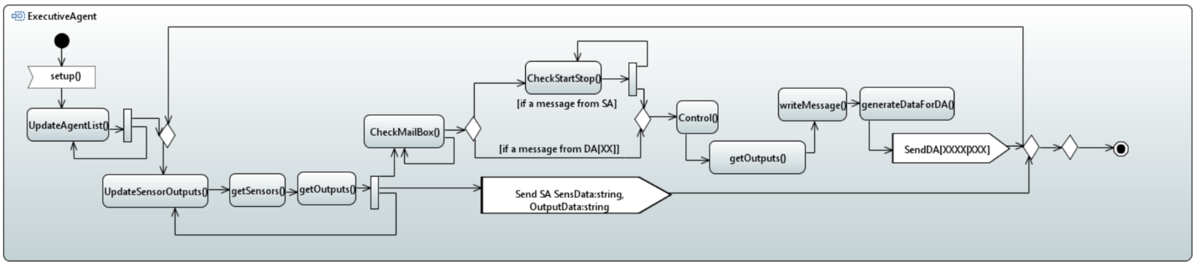


Figure 5.16TheActivity Diagram for Executive Agent

5.3.5 High-availability Agent

High-availability agent (HAA)was implemented providing two main functions: firstly, it checks the IP of agents, who registers in the system, so it will provide the possible attacks, and secondly, it provides Back-ups for agents, if they were terminated or failed.

Table 5.7 High-availability Agent main parameters and Behaviors

	Methods in a Class Diagram	Represents Activity
Behaviors in JADE	UpdateAgentList()	Update the available agents list
	restartExeAgent()	Restart EA if it was terminated or failed (back-up)
	restartDisAgent()	Restart DA if it was terminated or failed (back-up)
	restartReschAgent()	Restart RA if it was terminated or failed (back-up)
	restartSupAgent()	Restart SA if it was terminated or failed (back-up)
	writeMessage ()	Write a message to SA
	SearchInvalidIP()	Checking IP addresses (e.g.an attack)
	SearchForInvalidIP()	Triggers SearchInvalidIP()
Parameters (Properties in a Class Diagram)	*Private*ExeReset:boolean	Information about EA restart
	*Private*DispReset:boolean	Information about DA restart
	*Private*SupReset:boolean	Information about SA restart
	*Private*ReschReset:boolean	Information about RA restart

By HAA, the reliability of the multi-agent system is increased. If one Agent (SA, DA, EA or RA) turns out, it is started after a short time by HAA again. Furthermore, the HA checks the Internet address of each agent and ended each agent with a foreign Internet address.

Of course, there is always possibility to adjust the characteristics under customer's desires. Later, the main activities of the agent will be presented and shown on Activity Diagrams in Papyrus Software. The behaviors of the Agent are shown in the Table 5.7.

On the Figure 5.17 the presentation of the Rescheduler Agent in the view of a Class Diagram in Papyrus Software is shown.

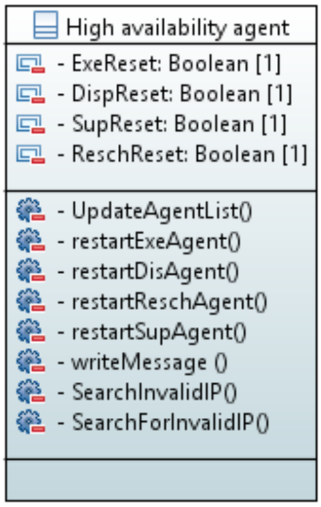


Figure 5.17The Class Diagram for HAA

For better describing of each implemented behavior, further for each behavior a table with sufficient information and ActivityDiagram is presented in the attachment.

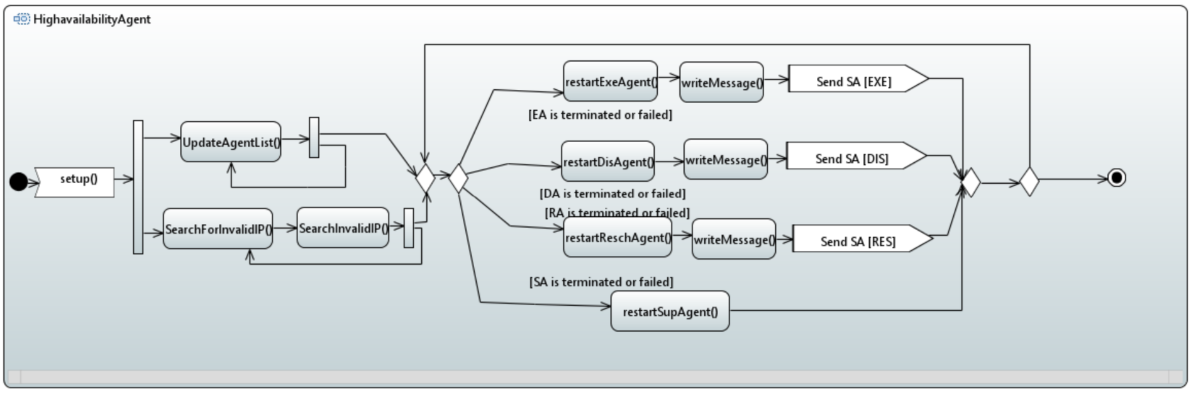


Figure 5.18TheActivity Diagram for High-availabilityAgent

Activity diagram presenting the general work of the High-availabilityAgentimplemented in JADE is showed on Fig.5.18.The modeling of all behaviors is presented in the attachment.

The general view of the interconnection between agents with general description is presented on Fig.5.19. On the Class Diagram is evident, that they are connected to each other and in the given system only one copy of the agent can exist.

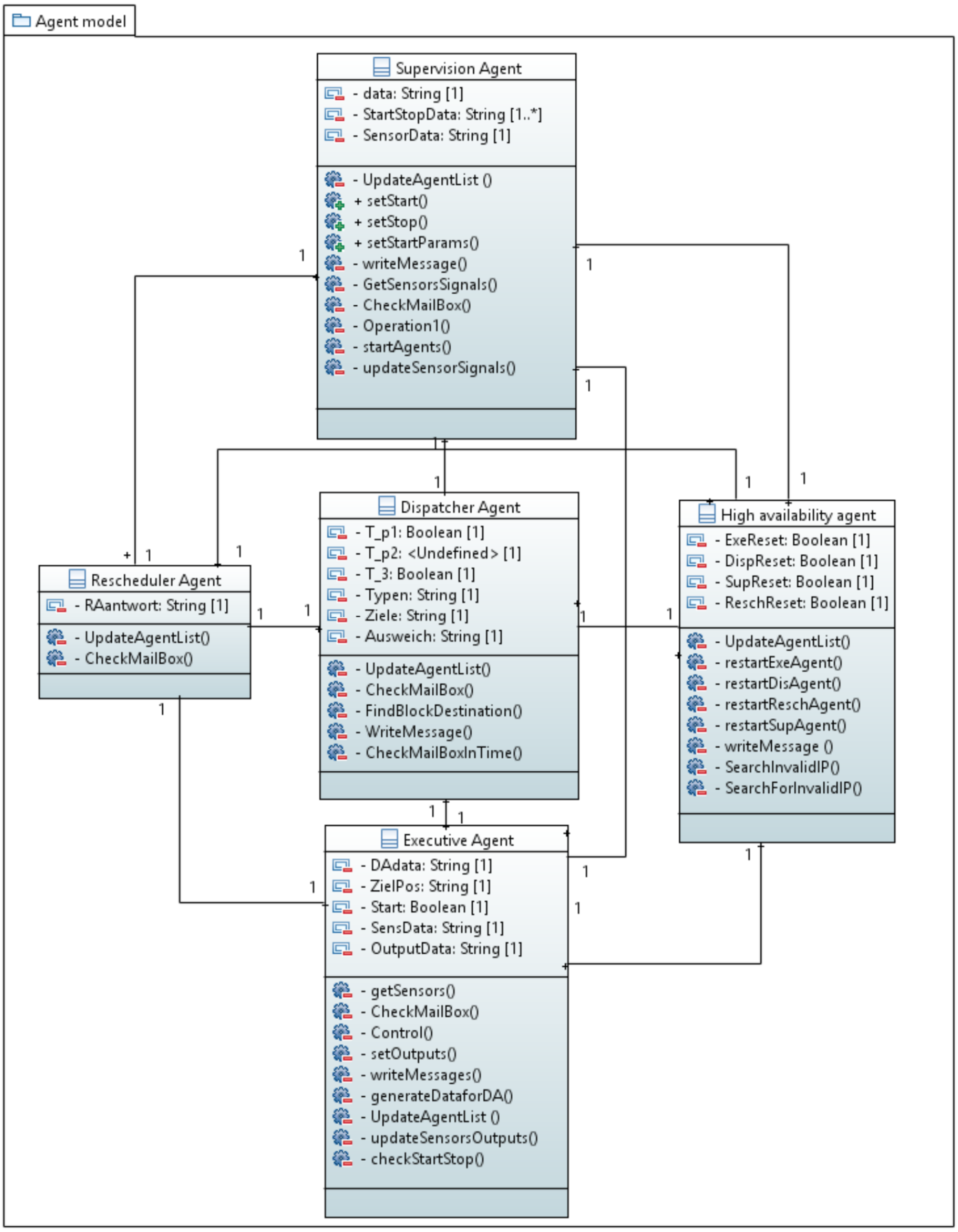


Figure 5.19 The general view of interconnection between all agents with general description on the Class Diagram

Summing up, in this Chapter, the full implementation of each agent was provided with all described behaviors. Any issues during the tests were not found. The cooperation of agents in the control architecture and availability of shown functions offers several advantages such as flexible variation of paths, a flexibility of hardware and software, and versatility of services.

6. Summary and outlook

6.1 Summary

To face the challenge of innovation, industry appeals to more smarterand self-adaptable manufacturing systems. This trend might suddenly increase risks, errors and costs, as well as the time, spent to conceive, develop and deliver a new product. A precise definition of the requirements, functions and architecture of a smart manufacturing system shows nowadays a need for a new universal control approach without big spending on time and money. This approach also should possess intelligence feature and be flexible to disturbances.

In the thesis, a precise description of the development of such an approachwas presented. The aim of the research was to analyze, how to move from legacy systems to a digital "smart" factory with intelligent filling, which will produce high customized products for consumer´s desires, which could flexibly deal with disruptions and failures and facilitate optimized decision-making possibly avoiding of human participation according to Industry 4.0 implementation recommendations.

During the development the following questions were delivered and answered:

Question 1. How to move from legacy systems to the intelligent and flexible control system?

This question was divided into two questions about the migration path from legacy control systems to the intelligent flexible control systems and about intelligence presence in the control system. Industrie 4.0 and CPS paradigms describe how the migration way from legacy systems to the intelligent control systemis possible. In the thesis,the answer on this question was shown on the proposed exampleof the Fischertechnik model using flexible multi-agent control systems on the additional Raspberry Pi controllers connected to Bus coupler and placed above the legacy control system in Chapter 5. The migration path was shown according to developing steps that were proposed in the thesis. After the carried out low-costimplementation, theobtained control system shows good results in performance.

The intelligence and the flexibility application in the industry allow receiving the versatility of products, mixes, volumes in the production. Hence, the second part of this question about intelligencewas answered by the consideration of different types of flexibility provided by the system (Paragraph 2.2) and knowledge databases presence(Paragraph 3.3). Different intelligence sources were analyzed, such as supervisory control, knowledge-based systems, and multi-agent systems. After the analysis of the advantages of intelligent control compare to traditional solutions, in the Paragraph 3.3.5 it was decided to realize the intellectual side through the multi-agent systems and supervision.

Question 2. How to provide production flexibility in the manufacturing environment?

The answer to this question was shown in Table 2.2, where the different flexibility types for each application case were considered according to the application cases: volume-, routing-, mix-, product-, process- and expansion flexibility in the conveyor; the product-, process-, volume- and expansion flexibility in heater; and machine-, production-, mix-, product- and process flexibility in the robot on shop floor.

The named flexibility is provided in the control system by using the agents that have different roles and tasks in the system and the supervisor. Due to theprovision of the flexibility types, the control system is ready for different changes and disturbances in the industrial environmentand is able to be adjusted.

Question3. How to develop the control system that provides flexible production?

To define the development steps of the flexible control system the different development standards were analyzed.

Starting from the plant planning phases that were described in VDI 5200, also VDI3695,VDI2206, VDI2243 were analyzed and compared. After it, the graphical representation of the scheme of the control system design was given in the Fig. 3.2. In the end following four phases were obtained: the phase before product development, phase of product development, phase of technical product development and phase of product implementation. The phase before product development consists of following steps: target planning, technology valuation, system requirements definition and software requirements definition. The phase of product development consists of productplanning and development and layout design. The next phase of technical product development consists of mechatronic- and electronic planning, programdevelopment and user interface development. The last phase consists of assembly, product operation monitoring and quality checking.

Due to the steps, given in the scheme, the development and implementation of the flexible control system were performed. Monitoring of the control system showed that no failures were found.

Question 4. Which functions should possess the control system to be called universal?

Answering this question, the analysis of main important features of the intelligent controller was provided (Fig.4.1). As the result, the following universal entities of the control system were obtained:

- an entity to interact with a legacy system,
- an entity that comprises a knowledge base,
- an entity for the data processing and
- acentral control systemwith a supervisor.

The control system should have its own knowledge database with the opportunity to change it and the possibility of the manual adjustment. Also, it should withstand and adjust itself to external disturbances.

The named entities were transformed into five main agents (Fig.4.2):

- Executive Agent applied to interact with legacy control systems,
- Supervisor Agent applied to perform supervisorytasks,
- High-availability Agent applied to deal with external disturbances,
- Rescheduler Agent applied to perform data processing within the process control,
- Dispatcher Agent is applied to deal with a knowledge base to ensure sustainable control.

The exact functions of each agent were defined according to DACS methodology. That was made for all three different application case described at the beginning of the thesis. In the result of DACS methodology application, it was shown that in all three cases the set of agents and their functions remain the same. This allows naming the proposed architecture “universal”. Hence, the universal control architecture can be applied to other production cases with a further adjustment to an object.

The multi-agent systemfor the application case “conveyor” was implemented on the JADE platform to control a small transporting conveyor from Fischertechnik in the laboratory of Otto-von-Guericke University. The full application of the control system to the “conveyor” case is shown in the thesis; the application to other two cases “robot” and “heater” were made particularly because of the lack an opportunity to apply them on the real object.

Summing up, theflexible approach of a universal control model for production solutions was proposedin the result of the research.

The obtained control system is Industrie 4.0 and CPS compliable (Table 2.3) andprovides following features:

- It has "intelligent filling", which is presented as with the presence of the knowledge-based system and also the supervisor entity applied in multi-agent systems,
- It providesdifferent types of flexibility,
- It possessesuniversal features and functions, aggregated from different application types, and due to that it can be applied to different application cases,
- It is Industry 4.0 and CPS compliable,
- It is provided by the low-cost change in construction.

This approach differs from existing traditional approaches,obtaining flexible, programmable and dynamic architecture, with providing the efficient control which is high-responsible to disturbances.

6.2 Outlook

After implementation, the control architecture shows a good response to the changing environment during the runtime demonstrating its flexibility,explicating knowledge about the dependencies between products and production systems and within.

The critical view on the work and the future work refer specifically to the developed software:

1) The obtained control system may be improved with a bigger and more detailed GUI showing more options to the operator, like presenting more information about conveyor components, also by adding more control options from GUI.

2) The obtained control system may be improved with HAA additional functions, for example, some security options, like a leakage protection provision.

3) The obtained control system may be improved with possible additional optimization of all application cases, for example,either on the conveyor band speed, or optimization of fuel

combustion process, oroptimization of the drilling process applying to the available resources. This complex mathematic will be placed in the Supervisor Agent.

4) Other agents´ functionality can also be supplemented further in the direction of Industrie 4.0. As it was planned, the Rescheduler Agent and Dispatcher Agent should work with different types of tables, possessing the different types of information about the system. In the given case, applied in JADE they have access to tables with the named block types and destinations of a block. In the complex Industrie 4.0 system, where all elements aware about each other, the table of resources and their services can be embedded in Dispatcher Agent.

5)Due to that fact, that in combustion process it is very hard to find a mathematical model for the process, and sometimes this model is presented by a fuzzy logic, the mathematic in the controller can be improved due to the usage of complex calculations within the fuzzy logic studying process. The “modes” of the heater that would be presented by a fuzzy logic situation model can be changed in these cases. This type of functionality would be placed in the Rescheduler Agent as the responsible for the implementation of new functioning “modes”.

6) The control system can be made practically usable for other application cases.

Summing up the results,the provided control architecture,compared with the current situation in production control, is able to presenta good robust controlworking with legacy production systems, making them flexible throughupdating software components of early used control system in the absence of the possibility of changing machine structure.

References

[ABB16] Koziolek H. (2013), “Towards the automation cloud: Architectural challenges for a novel Smart Ecosystem“, Available at: http://www.slideshare.net/hkoziolek/towards-the-automation-cloud-architectural-challenges-for-a-novel-smart-ecosystem (Accessed: 21 August 2016).

[Abr96] Abrial J. (1996), “Steam boiler control specification problem“, In Formal methods for Industrial Application Vol. 1165, Springer Berlin Heidelberg, pp. 500-509.

[ACA13] Acatech Deutsche Akademie der Technikwissenschaften e. V. (2013),“Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0“, Available at: https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf (Accessed: 21 August 2016).

[Adl88] Adler P.S. (1988), “Managing Flexible Automation“, California Management Review.Vol. 30(3), pp. 34-56, University of California Press Journals.

[Alk11] Alkhateeb, F. and Maghayreh, E.A. and Doush, I.A.(ed.), (2016), “Multi Agent Systems – Modeling“, Control, Programming, InTechOpen, Available at: http://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications.(Accessed 21 August 2016).

[Ant11] Antonova I. and Batchkova I. (2011), “Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications“, In Intelligent Systems, 2008.IS '08. 4th International IEEE Conference, InTech pp. 67-91.

[AUC16] The auto channel webpage. Available at: http://www.theautochannel.com/news/2008/05/14/086943.1-lg.jpg, (Accessed: 21 August 2016).

[Bah10] Bahrami H. and Evans S. (2010), “Super-Flexibility for Knowledge Enterprises: A Toolkit for Dynamic Adaptation“, 2nd edn, Springer Berlin Heidelberg.

[Bal01] Balzer H. (2001), “UML kompakt mit Checklisten“, Berlin Spektrum akademischer Verlag.

[Bar15] Barbosa J., Leitão P., Adam E. and Trentesaux D. (2015), “Behavioural validation of the ADACOR2 self-organized Holonic multi-agent manufacturing system“, in Industrial Applications of Holonic and Multi-Agent Systems, 7th International Conference, HoloMAS 2015, Valencia, Spain, September 2-3, 2015, Springer International Publishing, pp. 59–70.

[Bau14] Bauernhansl, T. and Hompel, M. and Vogel-Heuser, B. (ed.) (2014), “Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration“ Springer Fachmedien Wiesbaden.

[BBC13] Robertson C. (2013), “The best engineers come from Germany“, Available at: http://www.bbc.com/news/business-24131534 (Accessed: 21 August 2016).

[Bel07] Bellifemine F.L., Caire G. and Greenwood D. (2007), “Developing Multi-Agent Systems with Jade“, John Wiley & Sons.

[Bel09] Van Belle J., Saint Germain B., Verstraete P., Valckenaers P., Ali O., Van Brussel H. and Cattrysse D. (2009), "A Holonic chain conveyor control system: An application", in Holonic and Multi-Agent Systems for Manufacturing, Springer Berlin Heidelberg, pp. 234–243.

[Bey15] Beyer T., Jazdi N. and Göhner P. (2015), "Knowledge-based Planning and Adaptation of Industrial Automation Systems", In IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), September 2015, Luxemburg, IEEE.

[Bon98] Bongaerts L. (1998), "Integration of Scheduling and Control in Holonic Manufacturing Systems". Thesis at: Katholieke Universiteit Leuven, Belgium.

[Bro10] Broy M. (ed.) (2010), "Cyber-Physical Systems: Innovation Durch Software-Intensive Eingebettete Systeme", Springer Berlin Heidelberg, pp. 17-31.

[Bro13] Broy M. (2013),"Engineering Cyber-Physical systems: Challenges and foundations", in Complex Systems Design & Management: Proceedings of the Third International Conference on Complex Systems Design & Management CSD&M 2012, Springer Berlin Heidelberg, pp. 1–13.

[Bru98] Van Brussel H., Wyns J., Valckenaers P., Bongaerts L. and Peeters P. (1998), "Reference Architecture for Holonic Manufacturing Systems: PROSA", Comput. Ind. Amsterdam, The Netherlands, Vol. 37(3), Elsevier Science Publishers B. V., pp. 255-274.

[Bub16] Bubnicki Z., "Knowledge-based and earning control systems", Encyclopedia of Life Support Systems: Control systems (EOLSS), robotics and automation, Vol. XVII, Available at: http://www.eolss.net/sample-chapters/c18/E6-43-25-02.pdf (Accessed: 21 August 2016).

[Bus04] Bussmann S., Jennings N. and Wooldridge M. (2004), "Multiagent Systems for Manufacturing Control: A Design Methodology", Springer Berlin Heidelberg.

[Bus99] Bussmann S. and McFarlane D. (1999), "Rationales for holonic manufacturing control", In Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems, Belgium, Leuven, September, 1999.

[Cal16] Calà A., Ryashentseva D. and Lueder A. (2016), "Modeling Approach for a flexible manufacturing control system using SysML" (submitted), In IEEE 21th Conference on Emerging technologies and factory automation, September, 2016, Berlin, IEEE.

[Cap16] Investopedia.com (2003),"Capital intensive", in Available at: http://www.investopedia.com/terms/c/capitalintensive.asp (Accessed: 21 August 2016).

[Chr06] Chryssolouris G. (2006) Manufacturing systems: Theory and practice. 2nd edn, New York, NY : Springer Science+Business Media, Inc.

[Chr14] Chryssolouris G. and Wiendah H.P. (2014), "Factory", CIRP Encyclopedia of Production Engineering, Springer Berlin Heidelberg, pp. 500-503.

[Chr94] Christensen J. (1994), "Holonic Manufacturing systems: initial architecture and standards directions", In First European Conference on Holonic Manufacturing Systems, European HMS Consortium/University of Hanover, Hanover, Germany.

[CIP14] Laperrière L. and Reinhart G. (ed.) (2014), "CIRP encyclopedia of production engineering", Springer Berlin Heidelberg.

[CPS13] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA) (2013), “Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation, Technical report, April“.Available at: https://www.vdi.de/uploads/media/Stellungnahme_Cyber-Physical_Systems.pdf (Accessed: 21 August 2016).

[Cur06] Curry A., Flett P. and Hollingsworth I. (2006), “Managing Information & Systems: The Business Perspective“, London: Routledge Tailor and Francis Group.

[Das96] Das S. (1996), “The measurement of flexibility in manufacturing systems“, International Journal of Flexible Manufacturing Systems, Vol. 8(1), Kluwer Academic Publishers, pp. 67-93.

[Dav15] Davies R. (2015), “Industry 4.0: Digitalisation for productivity and growth“, Available at: http://www.europarl.europa.eu/RegData/etudes/BRIE/2015/568337/EPRS_BRI(2015)568337_EN.pdf (Accessed: 21 August 2016).

[Dem12] Demand metric (2012),“Product Development Process Diagram“,Available at: http://de.slideshare.net/demandmetric/product-development-process-diagram (Accessed: 21 August 2016).

[Die11] Diedrich C., Lüder A. and Hundt L. (2011), “Bedeutung der interoperabilität bei Entwurf und nutzung von automatisierten Produktionsystemen“, Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik., July, 2011. Vol. 59(7), Oldenbourg Wissenschaftsverlag, pp. 426-438.

[Dil91] Dilts D.M., Boyd N.P. and Whorms H.H. (1991), “The evolution of control architectures for automated manufacturing systems“, Journal of Manufacturing Systems.Vol. 10(1), Elsevier. pp. 79-93.

[Ecl16] Eclipse Foundation, Available at: http://www.eclipse.org/users/ (Accessed: 21 August 2016).

[ElM09] ElMaraghy H. (2009), “Changeable and Reconfigurable Manufacturing Systems“, Springer London.

[EuC09] European Commission (2009), ICT and Energy Efficiency, The Case for Manufacturing: Recommendations of the Consultation Group, Available at: ftp://ftp.cordis.europa.eu/pub/fp7/ict/docs/micro-nanosystems/smart-manufacturing_en.pdf (Accessed: 21 August 2016).

[Fac13] Foresight, Government Office for Science (2013), “The factory of the future“, Future of Manufacturing Project: Evidence Paper 29, Available at: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/283903/ep29-factory-of-the-future.pdf (Accessed: 21 August 2016).

[Far15] Farid A. (2015), “Designing Multi-agent Systems for Resilient Engineering Systems“, In Industrial Applications of Holonic and Multi-Agent Systems, Vol. 9266, Springer International Publishing, pp. 3-8.

[Fau15] Faul A. and Goehner P. (2015), “Framework für die dynamische kooperation von automatisierten Systemen“, Automation 2015.

[Fer11] Ferrarini L. and Lüder A. (2011) Agent-based technology manufacturing control systems, ISA, Available at: https://books.google.de/books?id=y89ZMwEACAAJ (Accessed: 21 August 2016).

[Fin14] Finaev V. (2014), “Prospects of the decision of the problem of management in the conditions of uncertainty“, “Izvestia SFedU. Engineering sciences“, 06, 2014, Vol. 2, Knowledge management in information systems, Available at: http://izv-tn.tti.sfedu.ru/?p=10493&lang=en (Accessed: 21 August 2016).

[FIP16] Huget M.P.(2016),“Welcome to the foundation for intelligent physical agents“, Available at: http://www.fipa.org (Accessed: 21 August 2016).

[Foe12] Foehr M., Leitao P., Wagner T., Jager T. and Luder A. (2012),“Integrating mechatronic thinking and multi-agent approaches“, Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), September 2012, IEEE.

[Fra13] Frank G., Westkämper E. and Schlögl W. (2013), “Smart Product Engineering“, Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, March 11th - 13th, 2013, Springer Berlin Heidelberg, pp. 613-622.

[Gen06] Gentil S. (2006), “Artificial Intelligence for Industrial Process Supervision“, In Proceedings on Advances in Applied Artificial Intelligence: 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2006, Annecy, France, Springer Berlin Heidelberg, pp. 2-11.

[Gir15] Giret A. and Trentesaux D. (2015), “Software Engineering Methods for Intelligent Manufacturing Systems: A Comparative Survey“, In Industrial Applications of Holonic and Multi-Agent Systems. Vol. 9266, Springer International Publishing, pp. 11-21.

[Goe13] Göhner P. (ed.) (2013), “Agentensysteme in der Automatisierungstechnik“, Springer Berlin Heidelberg.

[GRA16] Grace project (2013), Available at: http://grace-project.org/ (Accessed: 21 August 2016).

[Had05] Hadjiski M., Boshnakov K. and Christova N. (2005), “Multi-agent simulation in inference evaluation of steam boiler emission“, In 19th Eropean Conference on Modeling and Simulation, pp. 552-557.

[Har12] Hartung J., Baudzus B. and Deuse J. (2012),“A life cycle oriented evaluation of changeable manufacturing and assembly systems“, in Leveraging Technology for a Sustainable World: Proceedings of the 19th CIRP Conference on Life Cycle Engineering, University of California at Berkeley, USA, Springer Berlin Heidelberg, pp. 357–362.

[IND16] Industrial Internet consortium (2016), Available at: http://www.industrialinternetconsortium.org/ (Accessed: 21 August 2016).

[IOT16] FraunhoferGesellschaft (2016),“The Internet of things“, Available at: http://www.internet-der-dinge.de/en.html (Accessed: 21 August 2016).

[ISR16] VDI/VDE GMA, ZVEI (2016),“Industrie 4.0 – Technical Assets Basic terminology concepts, life cycles and administration models“, March, 2016,Available at: https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/6092_PUB_E_TW_GMA_Status_Report_ZVEI_-_Industrie_4_0_-_Technical_Assets_Internet.pdf (Accessed: 21 August 2016).

[Iva08] Ivan H. and Won P. (2008), “Multi-Agent Control System of a Kraft Recovery Boiler“, In 18th European Symposium on Computer Aided Process Engineering (ESCAPE 18), Elsevier.

[IWP14] Bauernhansl T., Diegner B. and Diemer J. (2014), “Industrie 4.0 - Whitepaper fuE-Themen “, April, 2014,Available at: http://www.ipa.fraunhofer.de/fileadmin/user_upload/Leitthemen/Industrie_4.0/Whitepaper_Plattform_Industrie_4.0.pdf (Accessed: 21 August 2016).

[Jen95] Jennings N. and Wooldridge M. (1995), “Applying Agent Technology“, Journal of Applied Artificial Intelligence, 9 special issue on Intelligent Agents and Multi-Agent Systems, pp. 351-361.

[Jim15] Jimenez J., Bekrar A. and Trentesaux D. (2015), “An Approach for Characterizing the Operating Modes in Dynamic Hybrid Control Architectures“, In Industrial Applicationsof Holonic and Multi-Agent Systems, HoloMAS 2015, Springer International Publishing Switzerland 2015, pp. 108-119.

[Kan14] Kanthabhabhajeya S. and Lennartson B. (2014), “Optimal performance of modular and synchronized mechatronic systems“, In IEEE 19th Conference on Emerging Technology and Factory Automation (ETFA), September 2014, IEEE , pp. 1-7.

[Kar15] Leitão, P. and Karnouskos, S. (ed.) (2015), “Agent-based control of production systems and its architectural challenges“, in “Industrial Agents: Emerging Applications of Software Agents in Industry“, Elsevier, pp. 153-170.

[Kar15a] Leitão, P. and Karnouskos, S.(ed.) (2015), Industrial Agents: Emerging Applications of Software Agents in Industry, Elsevier.

[Koc16] Koch, V. and Kuge, S. and Geissbauer, R. (2014), Strategy and formely Booz and Company: Industry 4.0 and Opportunities and challenges of the industrial internet, Available at: http://www.strategyand.pwc.com/media/file/Industry-4-0.pdf (Accessed: 21 August 2016).

[Kor06] Koren Y. (2006), “General RMS Characteristics. Comparison with Dedicated and Flexible Systems“, In Reconfigurable Manufacturing Systems and Transformable Factories, Springer Berlin Heidelberg, pp. 27-45.

[Kue15] Kühnle H. and Bitsch G. (2015), “Foundations & Principles of Distributed Manufacturing“, Springer International Publishing, Switzerland.

[KUK16] Automationworld – Covering the field of industrial automation, Available at: http://www.automationworld.com/sites/default/files/field/image/kukaspotwelding.jpg (Accessed: 21 August 2016).

[Lab16] Investopedia.com (2003),“Labor intensive“,Available at: http://www.investopedia.com/terms/l/laborintensive.asp (Accessed: 21 August 2016).

[Lee08] Lee E. (2008), “Cyber Physical Systems: Design Challenges“, Thesis at: Electrical Engineering and Computer Sciences University of California at Berkeley, January, 2008, (No. UCB/EECS-2008-8);Available at: http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html (Accessed: 21 August 2016).

[Leg11] Legat C., Neidig J. and Roshchin M. (2011), “Model-based Knowledge Extraction for Automated Monitoring and Control“, In Proceedings of the 18th IFAC World Congress, Milan.

[Leg13 Legat C., Lamparter S. and Vogel-Heuser B. (2013), “Knowledge-Based Technologies for Future Factory Engineering and Control“, In Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, Vol. 472, pp. 355-374, Springer Berlin, Heidelberg.

[Leg14] Legat C., Schütz D. and Vogel-Heuser B. (2014), “Automatic generation of field control strategies for supporting (re-)engineering of manufacturing systems“, Journal of Intelligent Manufacturing, October, 2014, Vol. 25, pp. 1101-1111.

[Lei04] Leitão P. (2004), “An Agile and Adaptive Holonic Architecture for Manufacturing Control“, Thesis at: Department of Electrotechnical Engineering Polytechnic Institute of Braganca, Portugal, Porto, Available at: http://www.ipb.pt/~pleitao (Accessed: 21 August 2016).

[Lei05] Leitão P. and Restivo F. (2005), “Experimental Validation of ADACOR Holonic Control System“, In HOLOMAS 2005, Vol. 3593, Springer Berlin, Heidelberg.

[Lei09] Leitão P. (2009), “Agent-based Distributed Manufacturing Control: A State-of-the-art Survey“, Eng. Appl. Artif. Intell. , Tarrytown, NY, USA, October, 2009, Vol. 22(7), pp. 979-991, Pergamon Press, Inc.

[Lei13] Leitão P., Marik V. and Vrba P. (2013), “Past, Present, and Future of Industrial Agent Applications“, IEEE Transactions on Industrial Informatics, Vol. 9(4), pp. 2360-2372.

[Lep15] Lepuschitz W., Lobato-Jimenez A., Axinia E. and Merdan M. (2015), “A Survey on Standards and Ontologies for Process Automation“, In Industrial Applications of Holonic and Multi-Agent Systems, Vol. 9266, Springer International Publishing, pp. 22-32.

[LeV14] Legat C. and Vogel-Heuser B. (2014), “Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics“, Springer International Publishing, pp. 195-208.

[Lue04] Lüder A., Klostermeyer A. and Peschke J. (2004), “Design Patterns for Distributed agent based factory automation“, In IMS International Forum - Global Challenges in Manufacturing, pp. 783-791.

[Lue10] Lüder A., Peschke J. and Sanz R. (2010), “Design Patterns for Distributed Control Applications“, In Distributed Manufacturing, Springer London, pp. 155-175.

[Lun12] Lunze J. (2012), “Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme“, Vol. 3, Auflage Oldenbourg Wissenschaftsverlag.

[MAN16] Manufuture-EU, Available at: http://www.manufuture.org/manufacturing/ (Accessed: 21 August 2016).

[Mar12] Marwala T. (2012), “Fuzzy Systems for Condition Monitoring“, In Condition Monitoring Using Computational Intelligence Methods, Springer London, pp. 131-150.

[Mar16] MarrB. (2016),“Why everyone must get ready for the 4th industrial revolution“, Available at: http://www.forbes.com/sites/bernardmarr/2016/04/05/why-everyone-must-get-ready-for-4th-industrial-revolution/#728baf9e79c9 (Accessed: 21 August 2016).

[Meh14] Mehta B. and Reddy Y. (2014), “Industrial process automation systems: Design and Implementation“, Elsevier Science,Available at: https://books.google.de/books?id=IzLAwAAQBAJ (Accessed: 21 August 2016).

[MIC16] “MadeinChina2025“(2016), Available at: https://www.csis.org/analysis/made-china-2025 (Accessed: 21 August 2016).

[Nan12] Nandi A. (2012), “GA-Fuzzy Approaches: Application to Modeling of Manufacturing Process“, In Statistical and Computational Techniques in Manufacturing, Springer Berlin Heidelberg, pp. 145-185.

[OxD16] Oxford (2016),“Oxford dictionaries“, in Oxford Dictionary, Available at: http://www.oxforddictionaries.com/ (Accessed: 21 August 2016).

[PAP16] Papyrus, Eclipse Model Development Tools (2016),Available at: http://www.eclipse.org/papyrus (Accessed: 21 August 2016).

[PHO16] PHOENIX CONTACTGmbH & Co. KG (2016), Available at: http://www.phoenixcontact.com/ (Accessed: 21 August 2016).

[Poe14] Pötter T., Folmer J. and Vogel-Heuser B. (2014), "Enabling Industrie 4.0 Chancen und Nutzen für die Prozessindustrie", In Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer Fachmedien Wiesbaden, pp. 159-171.

[Pri14] Priego R., Gangoiti U., Orive D. and Marcos M. (2014), "Agent-based reconfiguration at controller level", IEEE 19th Conference on Emerging Technology and Factory Automation (ETFA), September 2014, IEEE, pp. 1-4.

[RAM15] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA) (2015), Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), April, Available at: https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf (Accessed: 21 August 2016).

[Ram87] Ramadge P.J. and Wonham W.M. (1987), "Supervisory Control of a Class of Discrete Event Processes", SIAM J. Control Optim., Philadelphia, PA, USA, January, 1987, Vol. 25(1), Society for Industrial and Applied Mathematics, pp. 206-230.

[Rao92] Rao A. and Georgeff M. (1992), "An Abstract Architecture for Rational Agents", In Proceedings of the Third International Conference on Principles of Know ledge Representation and Reasoning, Morgan Kaufmann Publishers.

[Rod13] Rodrigues N., Perreira A. and Leitão P. (2013), "Adaptive Multi-Agent System for a Washing Machine Production Line", In Industrial Applications of Holonic and Multi-Agent Systems, 6th International Conference, HoloMAS 2013, Springer Berlin Heidelberg, pp. 212-223.

[Roe16] RöpkeH., HellK., ZawiszaJ., LüderA., Schmidt N. (2016), Identification of "Industrie 4.0" Component Hierarchy Layers, IEEE 21th International Conference on Emerging Technologies and Factory Automation (ETFA), September, 2016, Berlin, IEEE.

[Ros15] Rosendahl R., Schmidt N. and Lueder A. (2015), "Industry 4.0 value networks in legacy systems", IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), September, 2015, Luxemburg, IEEE.

[Rus95] Russell S.J. (1995), "Artificial Intelligence: A modern Approach", Alan Apt.,Available at: http://stpk.cs.rtu.lv/sites/all/files/stpk/materiali/MI/Artificial%20Intelligence%20A%20Modern%20Approach.pdf (Accessed: 21 August 2016).

[Rya15] Ryashentseva D., Lueder A. and Rosendahl R. (2015), "An optimization approach within supervisor architecture for boiler control", IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), September, 2015, Luxemburg, IEEE.

[San96] Sanchez A. (1996), "Supervisory control theory", In Formal Specification and Synthesis of Procedural Controllers for Process Systems, Vol. 212, Springer Berlin Heidelberg, pp. 71-83.

[Sau06] Sauer O. (2006), "Integriertes Leit- und Auswertesystem für Rohbau, Lackierung und Montage", Automatisierungstechnische Praxis, Vol. 48(10), pp. 38-43.

[Sch15] Schmidt N., Lueder A. and Rosendal R. (2015), "Surveying integration approaches for relevance in Cyber Physical Production Systems", IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), September, 2015, Luxemburg, IEEE.

[SCH16] Vacuum technology for efficient processes (2016), Available at: http://www.schmalz.com/ (Accessed: 21 August 2016).

[Sei13] Seixas I. and Leitão P. (2013), “Standards compliance in industrial agents applications“, In Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, Nov, 2013, pp. 7446-7451.

[SGA14] Group SGC (2014), “SG-CG/ M490/F_ Overview of SG-CG Methodologies“, Thesis at: CEN-CENELEC-ETSI Smart Grid Coordination Group, 11, 2014, (3), Available at: https://www.dke.de/de/std/informationssicherheit/documents/sgcg_methodology_overview.pdf (Accessed: 21 August 2016).

[SIE15] Siemens CT RTC SYE EMT-DE OMI (2015), “Advanced Manufacturing Process Chain: Closing the loop between shop-floor and CAD model (AMProC)“, Thesis at: Siemens CT RTC SYE EMT-DE, OvGU Magdeburg IAF.

[Sko14] Skolaut W. (ed.) (2014), “Maschinenbau: Ein Lehrbuch für das ganze Bachelor-Studium“, Springer Berlin Heidelberg.

[Sla83] Slack N. (1983), “Flexibility as a Manufacturing Objective“, International Journal of Operations and Production Management, In International journal of operations and production management, Vol. 3(3), Emerald backfiles, pp. 4-13.

[StR14] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA) (2014) Industrie 4.0 Statusreport Gegenstände, Entitäten, Komponenten, Available at: https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/sk_dateien/VDI_Industrie_4.0_Komponenten_2014.pdf (Accessed: 21 August 2016).

[StR15] VDI/VDE Society Measurement and Automatic Control (GMA) (2015) Status Report Reference Architecture Model Industrie 4.0 (RAMI4.0), Juli. Available at: http://www.zvei.org/Downloads/Automation/5305%20Publikation%20GMA%20Status%20Report%20ZVEI%20Reference%20Architecture%20Model.pdf (Accessed: 21 August 2016).

[Tas98] TassoC. and Arantes e OliveiraE.R.(ed.) (1998), “Development of Knowledge-Based Systems for Engineering“, Vol. 333, Springer Vienna.

[Ter09] Terkaj W., Tolio T. and Valente A. (2009), “Focused Flexibility in Production Systems“, In Changeable and Reconfigurable Manufacturing Systems, Springer London, pp. 47-66.

[Tol06] Tolio T. and Valente A. (2006), “An Approach to Design the Flexibility Degree in Flexible Manufacturing Systems“, In 16th Int. Conference on Flexible Automation and Intelligent Manufacturing, Ireland, June.

[Ton98] Toni A.D. and Tonchia S. (1998), “Manufacturing flexibility: A literature review“, International Journal of Production Research, Vol. 36(6), pp. 1587-1617.

[Tza89] Tzafestas S.G.(ed.)(1989), “Knowledge-Based System Diagnosis, Supervision, and Control“, Springer, US.

[Ule12] Ulewicz S., Schutz D. and Vogel-Heuser B. (2012), “Design, implementation and evaluation of a hybrid approach for software agents in automation“, IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA), 2012, IEEE.

[Ule14] Ulewicz S., Schutz D. and Vogel-Heuser B. (2014), “Integration of distributed hybrid multi-agent systems into an industrial IT environment: Improving interconnectivity of industrial IT systems to the shop floor“, In Industrial Informatics (INDIN), 2014 12th IEEE International Conference, July, 2014, pp. 519-525.

[Vis98] Viswanadham N. and Lakshminarayanan A. (1998), “An agent-based information architecture for shopfloor control“, Sadhana, Vol. 23, Part 4, August 1998, India, August, 1998, Vol. 23(4), pp. 359-376.

[Vog14] Vogel-Heuser B. (2014), “Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik“, In Industrie 4.0 in Produktion, Automatisierung und Logistik, Springer Fachmedien Wiesbaden, pp. 37-48.

[Vog15] Vogel-Heuser B., Schütz D. and Göhner P. (2015), “Agentenbasierte Kopplung von Produktionsanlagen“, in Informatik-Spektrum, Vol. 38(3), Springer Berlin, Heidelberg, pp. 191-198.

[Vrb13] Vrba P. (2013), “Review of Industrial Applications of Multi-agent Technologies“, In Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, Vol. 472, pp. 327-338, Springer Berlin, Heidelberg.

[Vrb14] Vrba P., Marik V., Pierluigi S., Leitão P., Zhabelova G. and Vyatkin V. (2014), “A Review of Agent and Service-oriented Concepts applied to Intelligent Energy Systems“, IEEE Transactions on Industrial Informatics, Vol. 10(3), pp. 1890-1903.

[Wan07] Wang, L. and Shen, W. (ed.) (2007), “Process Planning and Scheduling for Distributed Manufacturing“, SpringerLondon.

[Wei06] Weilkiens T. (2006), “Systems Engineering mit SysML /UML: Modellierung, Analyse, Design“, dpunkt Verlag.

[Wes07] Westkämper E. (2007), “Digital Enterprise Technology: Perspectives and Future Challenges“, Boston, MA, Springer US, pp. 3-14.

[Wey14] Weyrich M., Göhner P., Diedrich C., Fay A., Wollschlaeger M., Kowalewski S. and Vogel-Heuser B. (2014), “Flexibles Management einer dezentralen Automatisierungsverbundanlage als Beispiel für Industrie 4.0 - Aspekte der Systemmodellierung und dezentralen Architektur“, In AUTOMATION 2014 - Smart X - Powered by Automation, Baden-Baden, Germany, VDI/VDE GMA.

[Wie07] Wiendahl H.P., ElMaraghy H., Nyhuis P., Zäh M., Wiendah H.H., Duffie N. and Brieke M. (2007), “Changeable Manufacturing - Classification, Design and Operation“, CIRP Annals - Manufacturing Technology, Vol. 56, Springer Berlin Heidelberg, pp. 783-809.

[Wil93] Williams T. (1993), “The Purdue Enterprise Reference Architecture and Methodology (PERA)“, In Information Infrastructure Systems for Manufacturing, Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing, DIISM '93, Tokyo, Japan, 8-10 November, 1993.

.

Standards and Guidelines

[DIN62264-1] DIN (2013), "DIN EN 62264-1:2014 : Integration von Unternehmensführungs- und Leitsystemen", Beuth Verlag GmbH Berlin.

[DIN8580] DIN (2003), "DIN 8580 Fertigungsverfahren - Begriffe, Einteilung (Manufacturing processes - Terms and definitions)", Beuth Verlag GmbH Berlin.

[ISA95] ISA Committee of volunteer experts (2000), ANSI/ISA 95 Standard “Enterprise - Control System Integration Part 1: Models and Terminology”, Instrument Society of America.

[VDI2206] VDI (2004), "Richtlinie 2206 Entwicklungsmethodik fuer mechatronische Systeme", Beuth Verlag GmbH Berlin.

[VDI2221] VDI (1993), "Richtlinie VDI/VDE 2221 Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte", Beuth Verlag GmbH Berlin.

[VDI2243] VDI (2002), "Recyclingorientierte Produktentwicklung", Beuth Verlag GmBH Berlin.

[VDI2653-1] VDI/VDE (2010),“ VDI/VDE 2653 Blatt 1 Agentensysteme in der Automatisierungstechnik - Grundlagen, Multi agent systems in industrial automation – Fundamentals. Beuth Verlag GmbH Berlin.

[VDI2653-2] VDI/VDE (2012), "VDI/VDE 2653 Blatt 2 Agentensysteme in der Automatisierungstechnik - Entwicklung", Richtlinie. 01, 2012. Beuth Verlag GmbH Berlin

[VDI2653n] VDI/VDE (2016), "VDI/VDE 2653 Blatt 1 Agentensysteme in der Automatisierungstechnik: Grundlagen (Entwurf)", Beuth Verlag GmbH Berlin

[VDI3694] VDI/VDE (2014), "Richtlinie VDI/VDE 3694 Lastenheft/Pflichtenheft fuer den Einsatz von Automatisierungssystemen", Beuth Verlag GmbH Berlin.

[VDI3695] VDI/VDE (2010), "VDI/VDE 3695 Blatt 1 Engineering von Anlagen - Evaluieren und optimieren des Engineerings - Grundlagen und Vorgehensweise", Beuth Verlag GmbH Berlin.

[VDI3695a] VDI/VDE (2010), "VDI/VDE 3695 Blatt 2 Engineering von Anlagen - Evaluieren und optimieren des Engineerings - Themenfeld Prozesse", Beuth Verlag GmbH Berlin.

[VDI4499] VDI (2008), "VDI 4499 Blatt 1 Digitale Fabrik - Grundlagen", Beuth Verlag GmbH Berlin.

[VDI5200] VDI (2009), "Richtlinie VDI/VDE 5200 Blatt 1 Fabrikplannung Planungsvorgehen", Beuth Verlag GmbH Berlin.

[VDI5600] VDI (2007), " VDI 5600 Blatt 1 Fertigungsmanagementsysteme / manufacturing execution systems (MES)", Beuth Verlag GmbH Berlin.

[VDMA66421] VDMA 66421 (2012), Referenzprozess zur durchgängigen Produktionsplanung – Standardisiertes Vorgehen für das Engineering von Produktionssystemen, Beuth Verlag GmbH Berlin.

Appendix

Table 7.1 UpdateAgentList() behavior of Supervisor Agent

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
UpdateAgentList()	In the "setup()"	Ticker(3 sec)	+fe:FIPAException -allStarted:bool -HighAvailRestarted: bool	-DispatcherAge:AID -ExecuteAge:AID -ReschedulerAge:AID -HighAvailAge:AID -myGui:SupervisorGui	-Agents:AID[ ] +template: DFAgentDescription +sd:ServiceDescription +result: DFAgentDescription[ ] -HighAge:AID

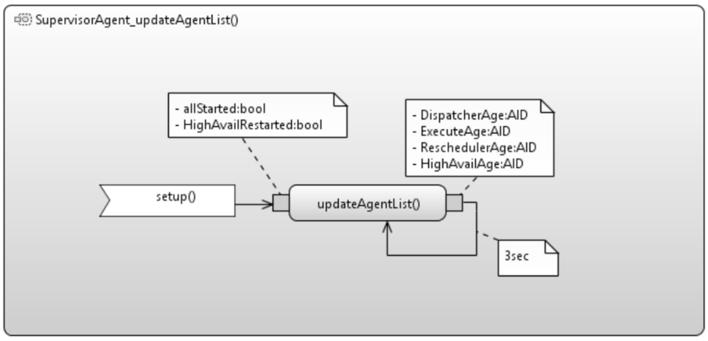


Figure 7.1 The Activity Diagram for UpdateAgentList()

Table 7.2 setStart() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
setStart()	"START" button is pressed	OneShot	none	-Start:bool -mStartStop:bool	-startStopData:String

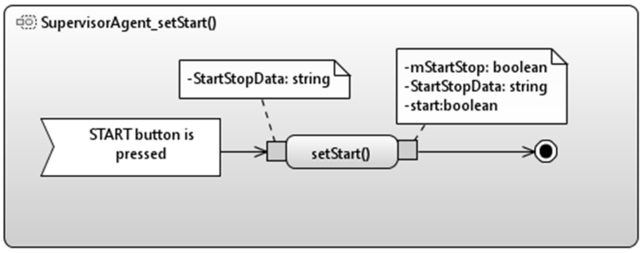


Figure 7.2 The Activity Diagram for setStart()

Table 7.3 setStop() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
setStop()	"STOP" button is pressed	OneShot	none	-Start:bool -mStartStop:bool	-startStopData:String

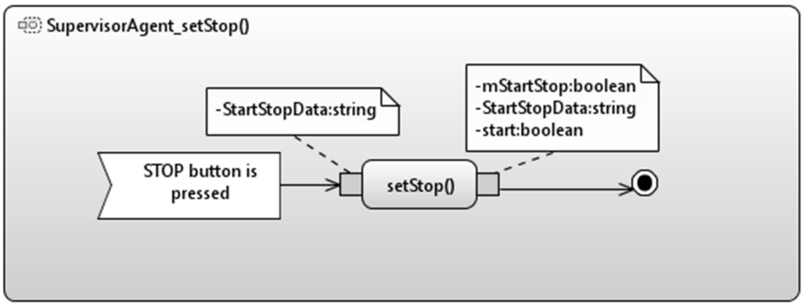


Figure 7.3 The Activity Diagram for setStop()

Table 7.4 setStartParams() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
setStartParams()	"ADD" button is pressed	OneShot	+tableP1:String +tableP2:String +table3:String	-data:String -temptischP1:String -temptischP2:String -temptisch3:String	-tischP1:String -tischP2:String -tisch3:String

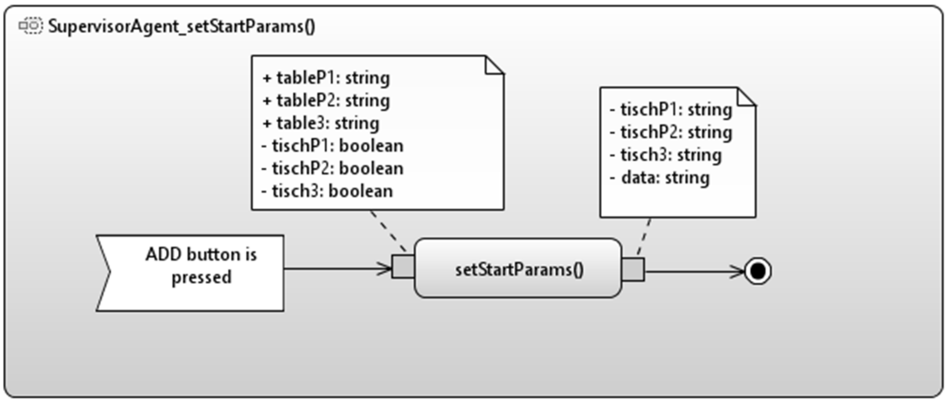
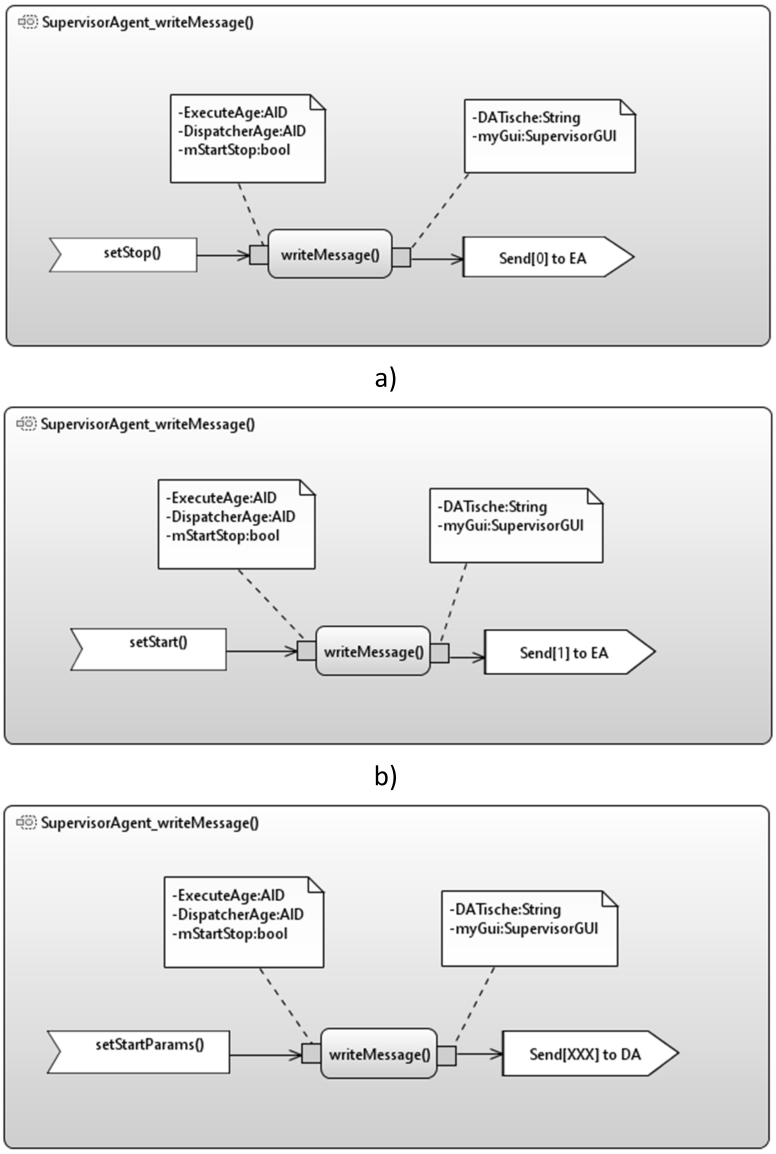


Figure 7.4 The Activity Diagram for setStartParams()

Table 7.5 writeMessage() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
writeMessage ()	After each "setStart ()", "SetStop ()" and "setStartParams ()"	special	-ExecuteAge:AID -DispatcherAge:AID -mStartStop:bool -startStopData:String -data:String	-DATische:String -myGui:SupervisorGUI	-step:int -mt:MessageTemplate -DisRestarted:bool -ExeRestarted:bool +cfp:ACLMessage +reply:ACLMessage



c)

Figure 7.5 The Activity Diagram for writeMessage ()

Table 7.6 getSensorSignals() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
getSensorSignals ()	After each "setStart ()", "SetStop ()" and "setStartParams ()"	Ticker (250 msec)	none	-SensorData:String -myGui:SupervisorGUI	-step:int -mt:MessageTemplate +cfp:ACLMessage +reply:ACLMessage

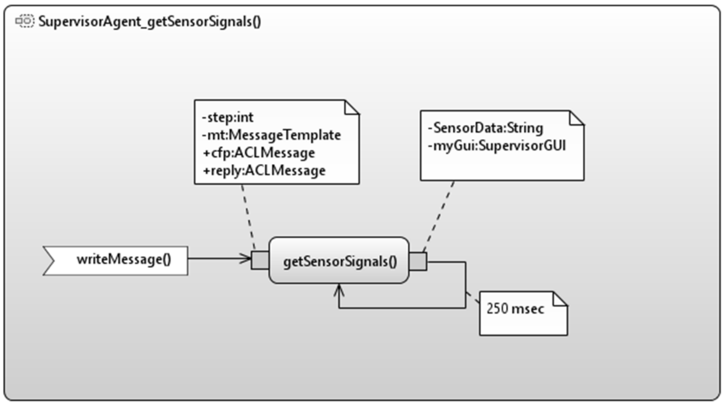


Figure 7.6 The Activity Diagram for getSensorSignals ()

Table 7.7 CheckMailBox() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
CheckMailBox()	In Setup() method	cyclic	-temptischP1:String -temptischP2:String -temptisch3:String	-ExeRestarted:bool -DisRestarted:bool -ResRestarted:bool -myGui:SupervisorGUI	+mt:MessageTemplate +msg:ACLMessage +title:String +reply:String

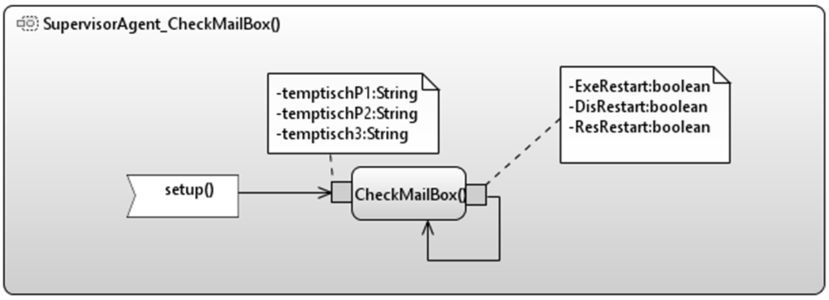


Figure 7.7 The Activity Diagram for CheckMailBox()

Table 7.8 updateSensorSignals() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
updateSensorSignals()	In „setup()“-Methode	Ticker 100ms	Triggers „getSensorSignals()“	none	none

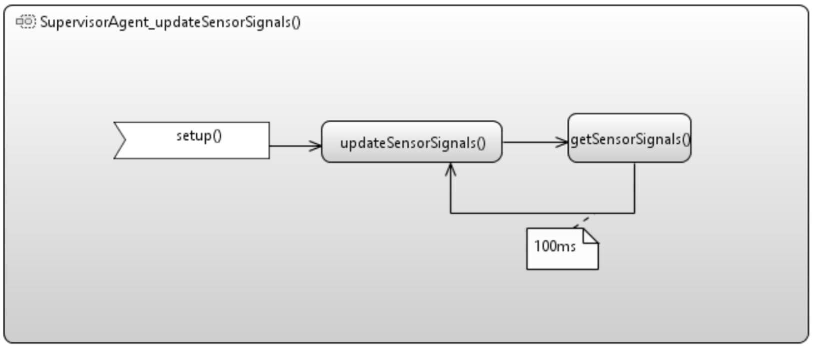


Figure 7.8 The Activity Diagram for updateSensorSignals()

Table 7.9 restartHighAvailAgent() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
restartHighAvailAgent ()	In „updateAgentList()“	OneShot	none	-HighAvailRestarted: bool	+mycontroller:AgentContainer +a:AgentController +e:StaleProxyException

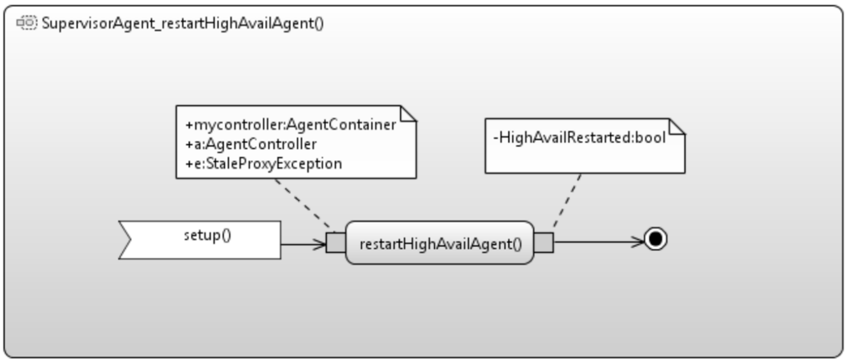


Figure 7.9The Activity Diagram Diagram for restartHighAvailAgent ()

Table 7.10 startAgents() behavior of SA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
startAgents ()	In „update AgentList()“	OneShot	none	-allStarted:bool	+mycontroller:AgentContainer +a:AgentController +b:AgentController +c:AgentController +d:AgentController +e:StaleProxyException

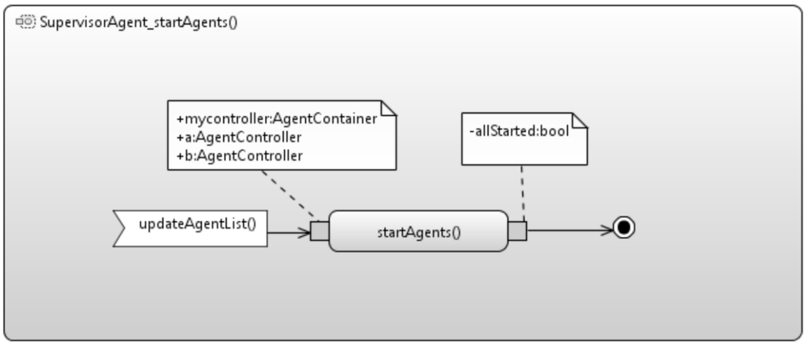


Figure 7.10 The Activity Diagram Diagram for startAgents()

Table 7.11 UpdateAgentList() behavior of Dispatcher Agent

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
UpdateAgent List()	In the "setup()" function	Ticker (3 sec)	+fe:FIPAException	-myGui:DispatcherGUI	-ExecuteAgent:AID -ReschedulerAgent:AID -SupervisorAgent:AID +template:DFAgentDescription +sd:ServiceDescription +result:DFAgentDescription[ ] -Agents:AID[ ]

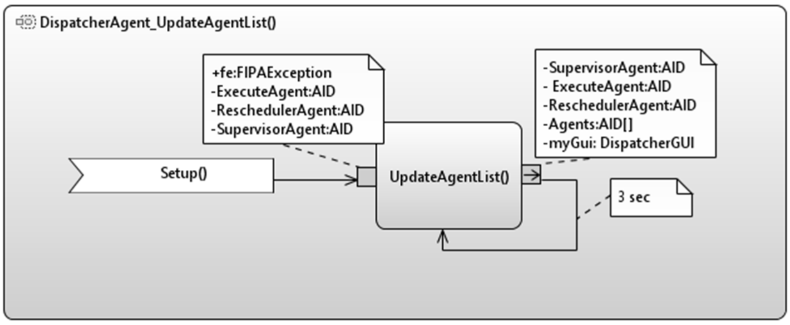


Figure 7.11 The Activity Diagram for UpdateAgentList()

Table 7.12 checkMailBox() behavior of DA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
checkMailBox()	In “setup()”	cyclic	-ExecuteAgent:AID -SuperviorAgent:AID	-Typ:String	+mt:MessageTemplate +msg:ACLMessage +title:String +reply:String

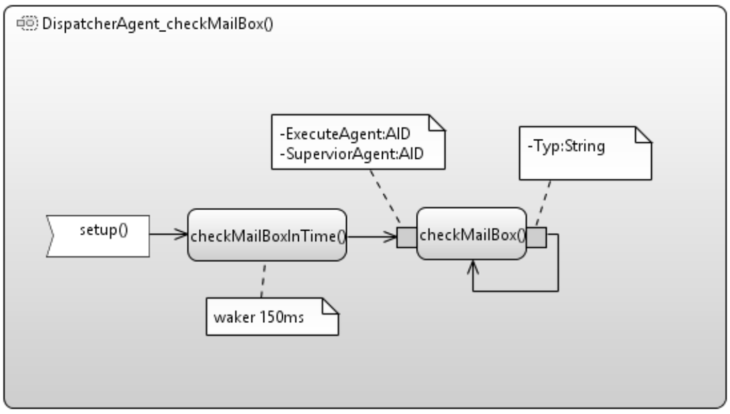
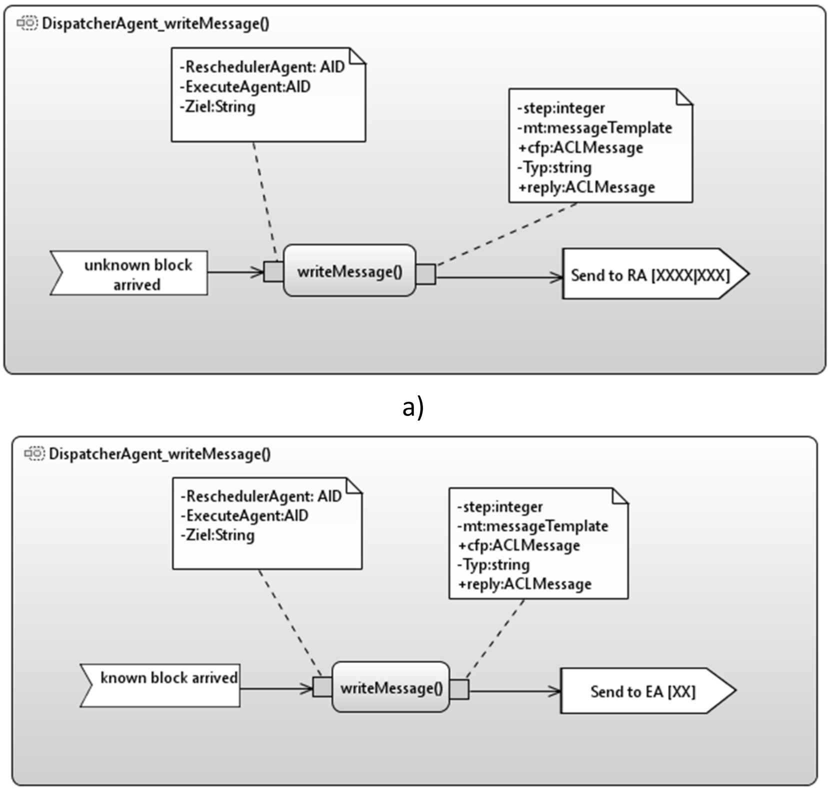


Figure 7.12 The Activity Diagram for checkMailBox()

Table 7.13 writeMessage() behavior of DA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name: type)
writeMessage()	In “FindBlockDestination()”If theunknown block arrived,then send a message toRA with template of magnets havebeen read and tables [XXXX_XXX];If the known block arrived, givethe path to EA [XX]	special	-ReschedulerAgent: AID -ExecuteAgent:AID -Ziel:String	none	-step:int -mt: MessageTemplate +cfp:ACLMessage +reply:ACLMessage -Typ:String



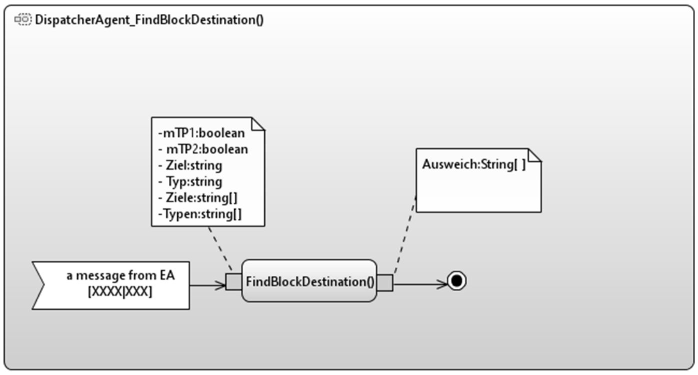
a)

b)

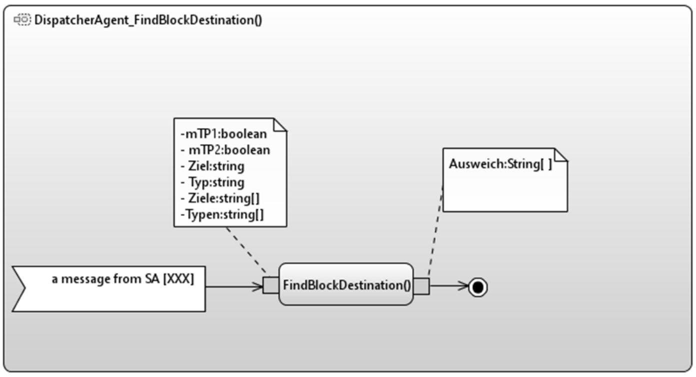
Figure 7.13 The Activity Diagram for writeMessage()

Table 7.14 FindBlockDestination()behavior of DA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
FindBlockDestination()	In “CheckMailBox”, if a message from EA or SA came	special	-Typ:string	-Ausweich:String[ ]	-mTP1:bool -mTP2:bool -mT3:bool -einmalnur:bool -einmalnur2:bool -BlockStep:int -Stelle:int -Ziel:String -Ziele:String[ ] -Typen:String[ ]



a)



b)

Figure 7.14 The Activity Diagram for FindBlockDestination()

Table 7.15 CheckMailBoxInTime() behavior of DA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
CheckMailBoxInTime ()	In “setup()”	Waker150ms	Triggers CheckMailBox() after 150ms wait	none	none

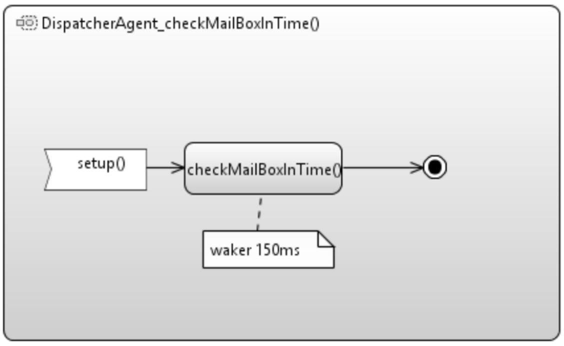


Figure 7.15The Activity Diagram for CheckMailBoxInTime ()

Table 7.16 UpdateAgentList() behavior of Rescheduler Agent

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
UpdateAgentList()	In the "setup()"	Ticker (3 sec)	+fe:FIPAException	-DispatcherAgent: AID	+template:DFAgentDescription +sd:ServiceDescription +result:DFAgentDescription[ ] -Agents:AID[ ]

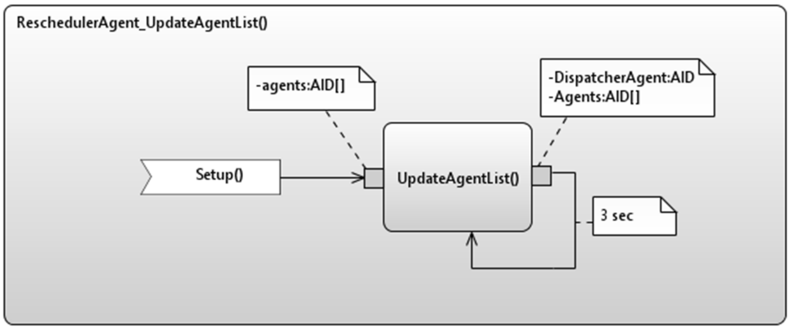


Figure 7.16 The Activity Diagram for UpdateAgentList()

Table 7.17 checkMailBox() behavior of RA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
checkMailBox()	In “setup()”	cyclic	-DispatcherAgent:AID -RAantwort:String	none	+mt:MessageTemplate +msg:ACLMessage +title:String +reply:String

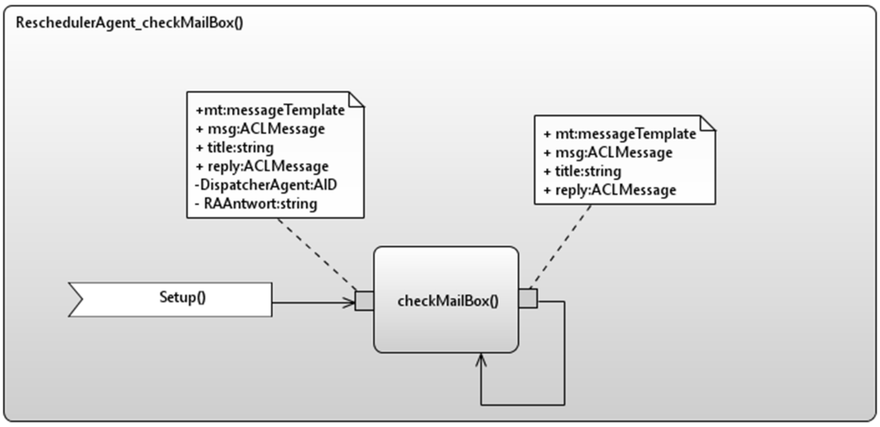


Figure 7.17 The Activity Diagram for checkMailBox()

Table 7.18 UpdateAgentList() behavior of Executive Agent

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
UpdateAgentList()	In the "setup()"	Ticker (3 sec)	+fe:FIPAException	-DispatcherAgent:AID	-Agents:AID[ ] +template:DFAgentDescription +sd:ServiceDescription +result:DFAgentDescription[ ]

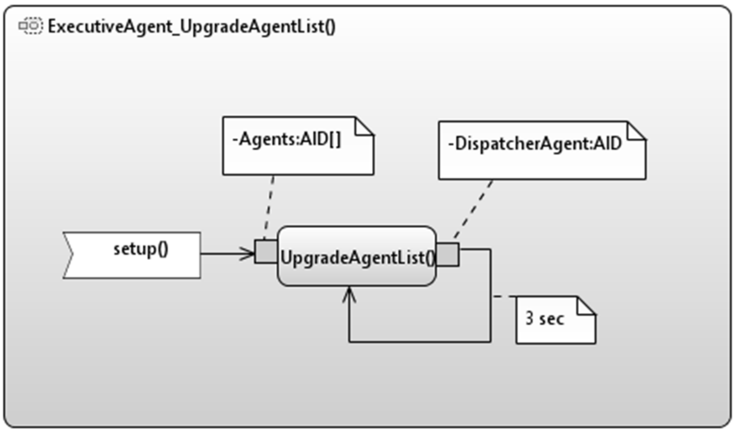


Figure 7.18 The Activity Diagram for UpdateAgentList()

Table 7.19 getSensors() behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name: type)
getSensors()	Actualization of sensor values	One shot	none	-Taster:bool -MagnetSP:bool -Magnet1:bool -Magnet2:bool -EndRausP1:bool -EndReinP1:bool -EndRausP2:bool -EndReinP2:bool -BlockVorP1:bool -BlockVorP2:bool -unused1:bool -unused2:bool -Licht:bool -TischP1:bool -TischP2:bool -Tisch3:bool	+arg:String[ ] -SensData:String

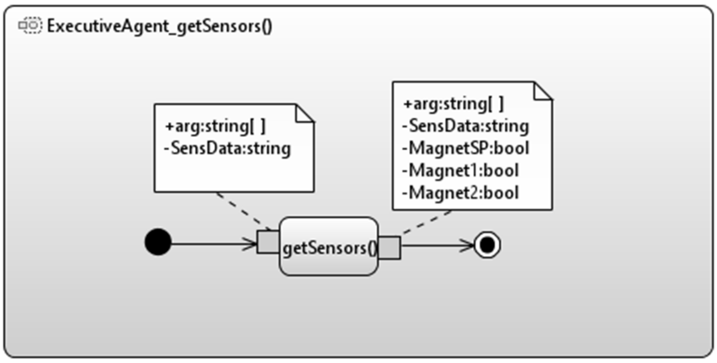


Figure 7.19 The Activity Diagram for getSensors() with Value “SensData” [XXXXXXXX_XXXXXXXX_]

Table 7.20 getOutputs() behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
getOutputs ()	In “setup()” function	Oneshot	none	-OutputData:String	+arg:String[ ]

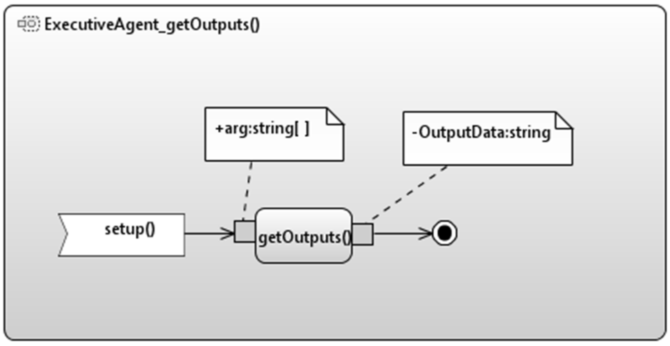


Figure 7.20 The Activity Diagram for getOutputs ()

Table 7.21 updateSensorsOutputs() behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
updateSensorsOutputs()	In “setup()”	Ticker (5 ms)	+ Triggers „getSensors“ und „getOutputs“		

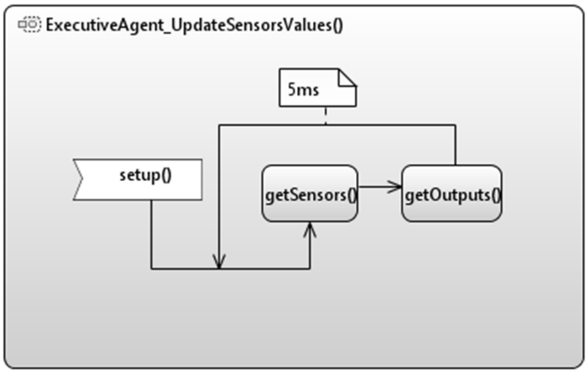


Figure 7.21The Activity Diagram for updateSensorsOutputs

Table 7.22 checkStartStop()behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
checkStartStop()	In “setup()”	Ticker (100 ms)	-start:bool	none	-merkerAN:bool -einmal2:bool

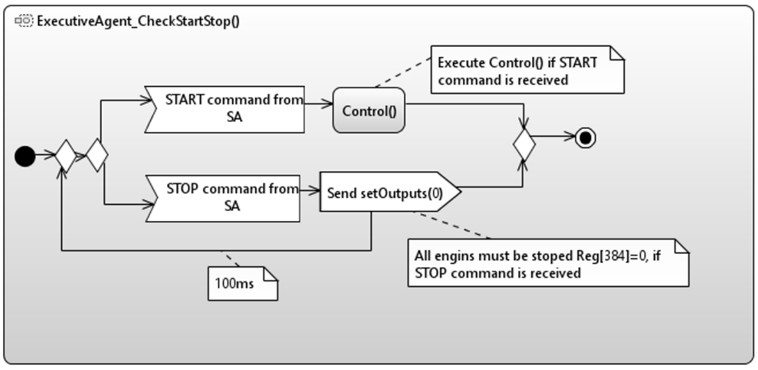


Figure 7.22 The Activity Diagram for checkStartStop()

Table 7.23 CheckMailBox()behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
CheckMailBox()	In “setup()”	Cyclic	-SensData:String -OutputData:String	-ZielPos:String	+mt:MessageTemplate +msg:ACLMessage +title:String +reply:String -start:bool

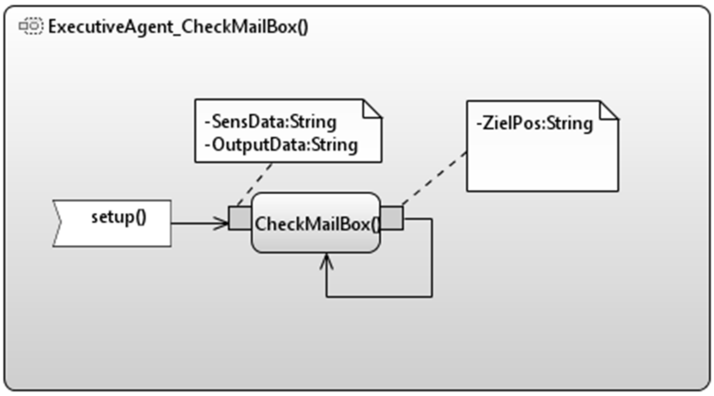


Figure 7.23 The Activity Diagram for CheckMailBox()

Table 7.24 Control()behaviorof EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
Control()	In VerificationStartStop()	Special	-start:bool -EndReinP1:bool -EndReinP2:bool -Licht:bool -ZielPos:String -MagnetSP:bool -Magnet1:bool -Magnet2:bool -Speicherband:int -P1Raus:int -P1Rein:int -P2Raus:int -P2Rein:int -GroßesBand:int -BlockVorP1:bool -BlockVorP2:bool -Tisch3:bool -TischP1:bool -TischP2:bool	none	-step:int -merkerL:bool -merkerP1raus:bool -merkerP2raus:bool -merker0:bool -merkerMagnetSP:bool -merkerMagnetSP2:bool -merkerGrBand:bool -merkerGefahren:bool -merkerM1ausgeloest:bool -merkerM1ausgeloest2:bool -merkerM1:bool -erstAusloeserM1:bool -zweitAusloeserM1:bool -merkerM1ausgeloest:bool -merkerM1ausgeloest2:bool -merkerM1:bool -erstAusloeserM1:bool -zweitAusloeserM1:bool -beginneP1fahren:bool -beginneP2fahren:bool -merkerP1raus:bool -merkerP2raus:bool

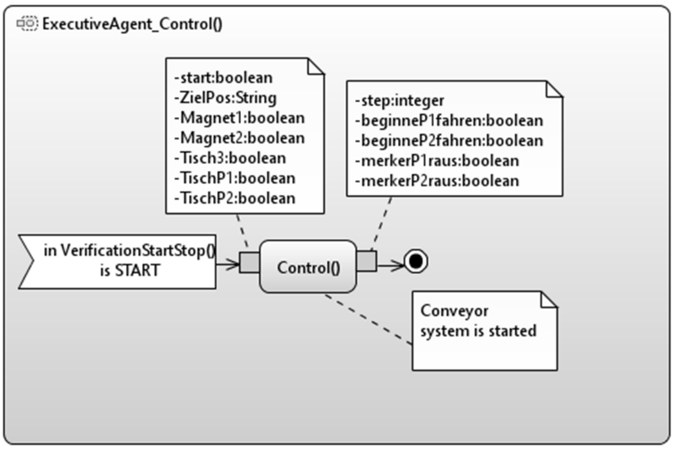


Figure 7.24 The Activity Diagram for Control()

Table 7.25 setOutputs(intval)behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
setOutputs(intval)	in „Control()“-Behavior	OneShot	-SensData:String -OutputData:String	-ZielPos:String	+val:int +argus:String[ ] -start:bool

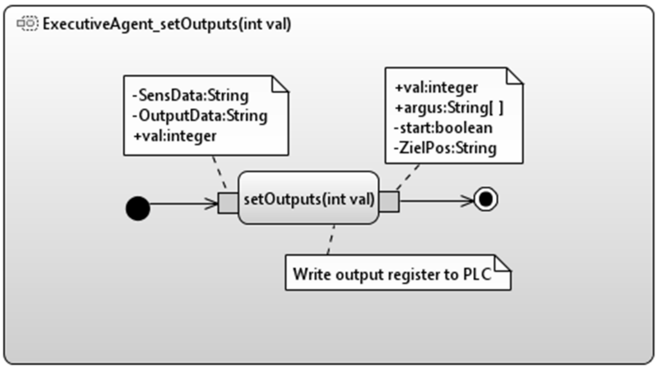


Figure 7.25 The Activity Diagram forsetOutputs(intval)

Table 7.26 writeMessage()behavior of EA

Behavior	Time of execution	Type of Beha	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
writeMessage()	in „Controll()“-Behavior	Special	-DispatcherAgent:AID -DAdata:String	+cfp:ACLMessage -merkerM1:bool -merkerM2:bool -erstAusloeserM1:bool -zweitAusloeserM1:bool -erstAusloeserM2:bool -zweitAusloeserM2:bool	-step:int -mt:MessageTemplate -einmal:bool -DAdataready:bool +reply:ACLMessage

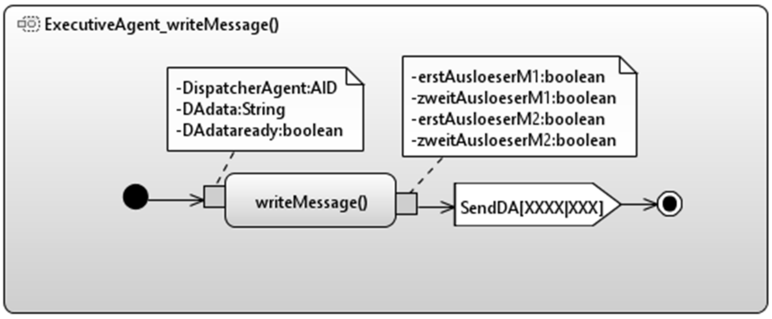


Figure 7.26 The Activity Diagram for writeMessage ()

Table 7.27 generateDataForDA()behavior of EA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
generateDataForDA()	in „writeMessage()“-Behavior	OneShot	-SensData:String -erstAusloeserM1:bool -zweitAusloeserM1:bool -erstAusloeserM2:bool -zweitAusloeserM2:bool -Magnet1:bool -Magnet2:bool -TischP1:bool -TischP2:bool -Tisch3:bool	-DAdataready:bool -ZielPos:String	-DAdata:String

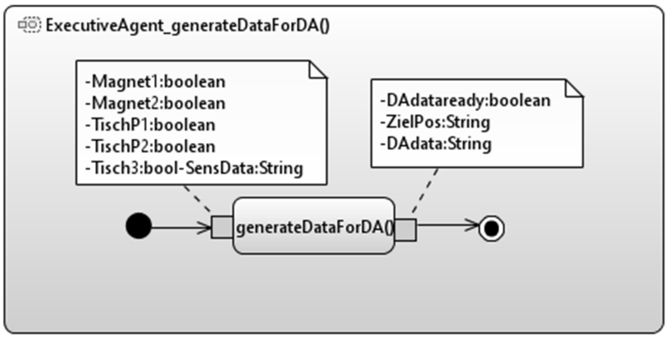


Figure 7.27 The Activity Diagram for generateDataForDA()

Table 7.28 UpdateAgentList() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
UpdateAgent List()	In the "setup()"	Ticker (500ms)	none	-ExeAge:AID -DisAge:AID -ReschAge:AID -SupAge:AID	*private*Agents:AID[] -ExecuteAge:AID -DispatcherAge:AID -ReschedulerAge:AID -SupervisorAge:AID

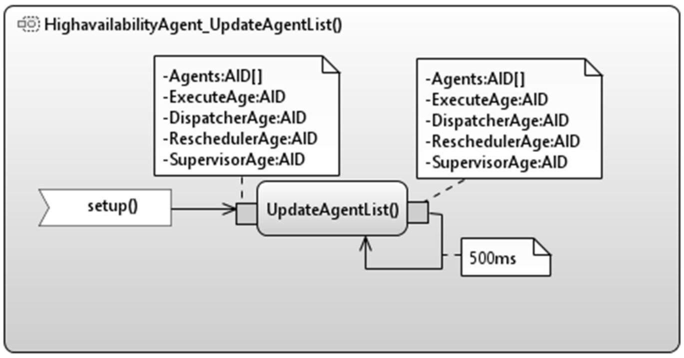


Figure 7.28 The Activity Diagram for UpdateAgentList()

Table 7.29 restartExeAgent() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
restartExeAgent ()	In "setup()"	Ticker (500ms)	none	-ExeReset:boolean	+myController: Agent Container +a:Agent Controller

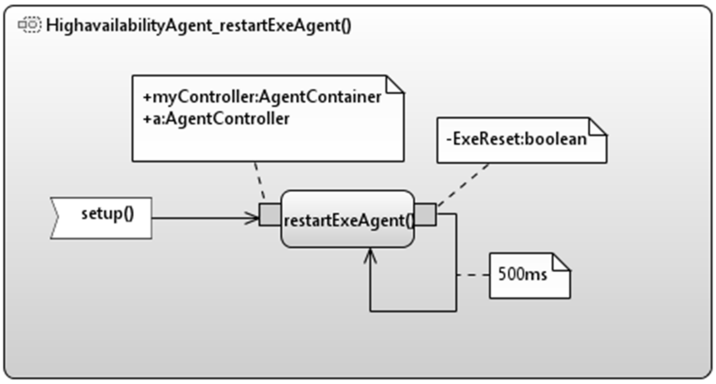


Figure 7.29 The Activity Diagram for restartExeAgent()

Table 7.30 restartDisAgent() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name: type)	Inout (*visibility*_name:type)
restartDisAgent ()	In "setup()"	Ticker (500ms)	none	-DispReset:boolean	+myController: Agent Container +a:Agent Controller

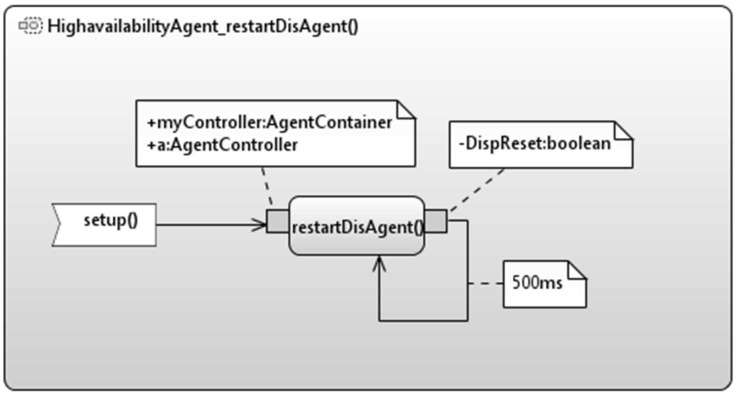


Figure 7.30 The Activity Diagram for restartDisAgent()

Table 7.31 restartReschAgent() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name:type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
restartResch Agent ()	In “setup()”	Ticker (500ms)	none	-ReschReset:boolean	+myController: Agent Container +a:Agent Controller

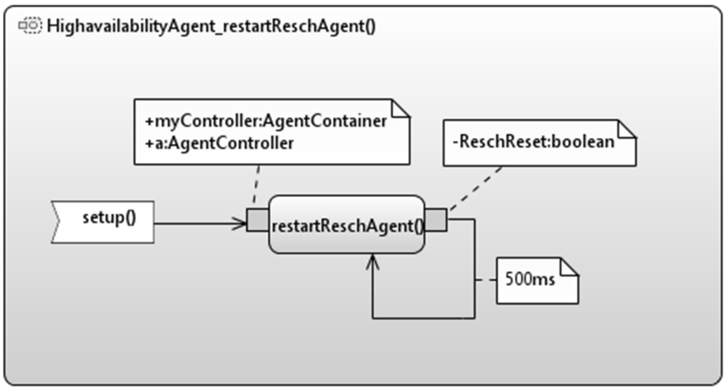


Figure 7.31 The Activity Diagram for restartReschAgent()

Table 7.32 restartSupAgent() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
restartSup Agent ()	In “setup()”	Ticker (500ms)	none	-SupReset:boolean	+myController: Agent Container +a:Agent Controller

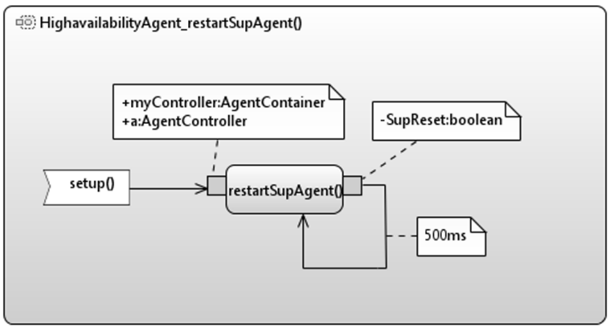


Figure 7.32 The Activity Diagram for restartSupAgent()

Table 7.33 writeMessage() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
writeMessage ()	After restart of RA/DA/EA	special	-SupAge:AID	none	-step:integer -mt:message template -ExeReset:Boolean -DispReset:Boolean -SupReset:Boolean -ReschReset:Boolean +cfp:ACLMessage +reply:ACLMessage

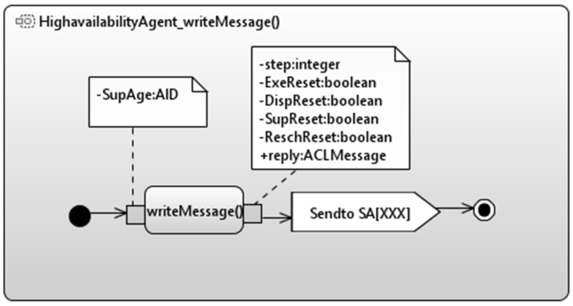


Figure 7.33 The Activity Diagram for writeMessage()

Table 7.34 SearchForInvalidIP() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
SearchFor InvalidIP()	In setup()	Ticker 3 sec	Triggers „SearchInvalidIP ()	none	none

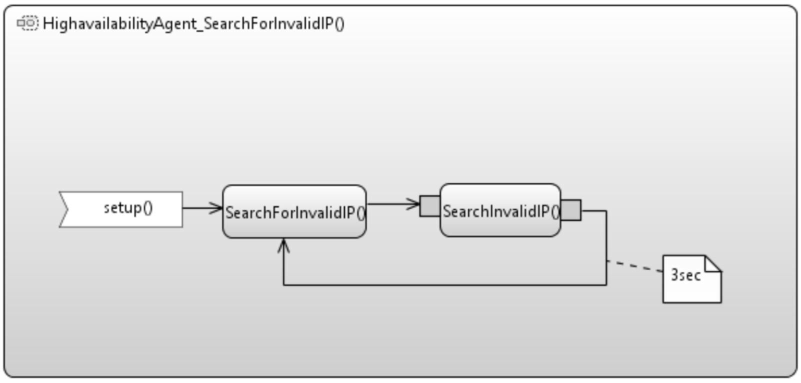


Figure 7.34 The Activity Diagram for SearchForInvalidIP()

Table 7.35SearchInvalidIP() behavior of HAA

Behavior	Time of execution	Type of Behaviors	In (*visibility*_name: type)	Out (*visibility*_name:type)	Inout (*visibility*_name:type)
SearchInvalidIP ()	In „SearchForInvalidIP()“	OneShot	-Agents:AID[ ]	none	-address:String[ ] +i:int +j:int

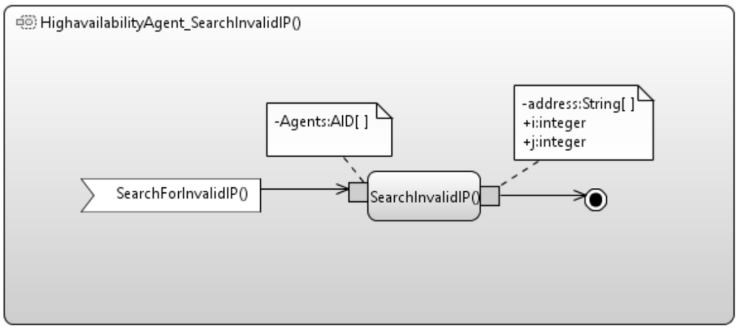


Figure 7.35The Activity Diagram for SearchInvalidIP ()