“LUCIAN BLAGA“ UNIVERSITY OF SIBIU
ENGINEERING FACULTY

PHD THESIS (SUMMARY)

RESEARCH REGARDING THE DIGITAL FACTORY MODELLING AND IMPLEMENTATION INTO REAL MANUFACTURING SYSTEMS

Scientific advisor:
Prof. Dr. Ing. Carmen SIMION

PhD student:
Dipl. Ing. Bogdan-Constantin PÎRVU

Sibiu
-2011-

UNIUNEA EUROPEANĂ
GUVERNUL ROMÂNIEI MINISTERUL MUNCII, FAMILIEI ŞI PROTECŢIEI SOCIALE AMPOSDRU
Fondul Social European POS DRU 2007-2013
Instrumente Structurale 2007-2013
MINISTERUL EDUCAŢIEI CERCETĂRII TINERETULUI ŞI SPORTULUI OIPOSDRU
Universitatea Lucian Blaga Sibiu

Investeşte în oameni!
PROIECT FINANŢAT DIN FONDUL SOCIAL EUROPEAN
ID proiect: 7706
Titlul proiectului: „Creşterea rolului studiilor doctorale şi a competitivităţii doctoranzilor într-o Europă unită"
Universitatea"Lucian Blaga" din Sibiu
B-dul Victoriei, nr. 10. Sibiu

"Lucian Blaga" University of Sibiu
Engineering Faculty

PhD thesis (Summary)

RESEARCH REGARDING THE DIGITAL FACTORY MODELLING AND IMPLEMENTATION INTO REAL MANUFACTURING SYSTEMS

Scientific advisor:
Prof. Dr. Ing. Carmen SIMION

PhD student:
Dipl. Ing. Bogdan-Constantin PÎRVU

Sibiu, 2011

Ministry of Education, Research, Youth and Sport
“Lucian Blaga” University of Sibiu
B-dul Victoriei Nr. 10, 555024 – Sibiu, Romania

COMPETENCE
PhD commission, named by the “Lucian Blaga” University of Sibiu
rector order
nr. 103 from 30th September 2011

President:	Prof.Dr.-Ing. Ioan Bandrea
	“Lucian Blaga” University of Sibiu
Scientific advisor:	Prof.Dr.-Ing. Carmen Simion
	“Lucian Blaga” University of Sibiu
Referents:	Prof.Dr.-Ing. George-Emilian Drăghici
	“Politehnica” University of Timisoara
	Prof.Dr.-Ing. Nouraş-Barbu Lupulescu
	Transilvania University of Braşov
	Prof.Dr.-Ing. Daniel Volovici
	“Lucian Blaga” University of Sibiu

Please send appreciations and observations to the “Lucian Blaga” University of Sibiu address

Preface

The PhD thesis was developed based on the research performed in laboratories of the "Hermann Oberth" Engineering Faculty within the "Lucian Blaga" University of Sibiu, but also during the research period that took place at the University of Kaiserslautern and at the innovative research and demonstration platform SmartFactory^{KL}.

At the end of this phase, my thoughts and gratitude are turning towards the ones hat supported me to elaborate and complete my PhD thesis.

On this occasion, I would like to express my thanks to Professor Dr. Eng. Carmen Simion, the scientific advisor of my PhD thesis, for the competence with which she coordinated the entire activities undertaken during the PhD thesis preparation and the full support offered to overcome unforeseen obstacles. I would also like to thank Professor Dr. Eng Ioan Bondrea for the initiation, support and coordination offered before and during the PhD thesis preparation.

Special thanks to Dr. Eng. Jochen Schlick, head of the production automation research department within the University of Kaiserslautern, for the support and the scientific and moral expertise, the knowledge accumulated being fundamental in the PhD thesis preparation and completion.

I would also like to thank Professor Dr. Eng. Detlef Zühlke, the chairman and initiator of the research and demonstration platform SmartFactory^{KL} and also head of the production automation department within the University of Kaiserslautern, for the permission to conduct research at the highest level.

I would like to thank and show my gratitude and honor to the PhD thesis commission members not only for the evaluation, support and thesis analysis but also for the permission to participate at the commission works.

Also, I would like to thank all the colleagues from the Sibiu Engineering Faculty, especially the ones from the Mechanical and Computer-science department, which offered their support and advice for the thesis completion.

More than gratitude, I express especially to my parents and close ones, which during my evolution have contributed to my upbringing, to the man I have become.

September, 2011 Bogdan-Constantin Pîrvu

Contents

* PhD thesis pages
** PhD thesis summary pages

1. Introduction

Globalization brings with it new challenges that the companies must overcome. The global market imposes the global competition where the main focus is the global client. The client (independent of its location) wants technical advanced products characterized by superior quality, high level of customization, low price that should be available at the right time and place. The problem is more complex due to the high fluctuation level of the customers' demands.

Making a parallel with Charles Darwin's theory of evolution, the companies which will survive and develop will not be the strongest nor the smartest but the ones better adapted to change. Adaptability and agility are the essential characteristics that allow a company to survive and flourish in the current turbulent and uncertain global environment.

Different concepts strive to offer increased company agility. If at the manufacturing and execution level are developed flexible or reconfigurable manufacturing systems, an integrative concept, relatively new, by the name of Digital Factory, promises to offer improved reactivity and agility. The main idea is to build in simulation (focuses mainly manufacturing processes) the "image" of the current and future factory, to analyze its parameters and to transfer the optimum manufacturing scenario to the real manufacturing system. One of the most important aspects that this concept implies is the capacity to build with ease and precision the manufacturing processes (current or future) into simulation, and, to transfer the obtained data from the simulation to the controller level respectively to the real manufacturing system.

Thus, manufacturing simulation is currently used for: manufacturing systems productivity increase, cost reduction for the manufacturing systems introduction, consumption optimization and resources reutilization, stock and manufacturing reduction. The major informational bottleneck and thus the main fundamental implementation issue of the Digital Factory concept takes place when the optimized manufacturing scenario from the simulation is transferred to the manufacturing system. Such a data migration, from the simulation level (in the PhD thesis are focused material flow and DMU - Digital Mock-Up) to the real manufacturing systems, in most of the cases is not foreseen or implies great efforts, interpretation or adaptation at the control unit (e.g. PLC).

To support the Digital Factory concept implementation, from the manufacturing automation standpoint new standards and approaches were researched, developed and introduced that allow for a distributed control of the whole manufacturing process (e.g. IEC 61499), thus, the manufacturing reconfiguration can be faster done without the "adaptation" of the central control unit (traditional scenario). Also, for the support of the concept implementation, an improved interaction between applications involved either at the command, planning, visualization, supervisory or at the simulation level is possible with the development of the

OPC UA specification. OPC UA allows the information transfer by a standardized or particular manner corresponding to different use-cases, which offers at the same time independence from any specific platform (e.g. Windows, Linux, Mac OS etc.) on which various applications are installed on.

1.1 Research motivation

A central research topic within the industrial engineering targets the agility improvement of manufacturing systems in order to allow a quick reaction to the global market variation. If at the hardware and software level, neutral and distributed controlling solutions are researched (e.g. IEC 61499 or the SOA approach for automation- implemented mainly based on web standards), at the manufacturing simulation level there are numerous solutions, which in most cases are offering just limited information migration towards the control units of manufacturing systems.

Thus, the informational bottleneck, located at the border of simulation application and the control unit of the real manufacturing system, represents a major impediment from a fast and efficient implementation of an optimum manufacturing scenario. Currently there are necessary several adaptation operations which are difficult, time-consuming and not error-proof in order to implement an optimum manufacturing scenario determined within the simulation.

The PhD thesis proposes a new simulation architecture regarding the manufacturing simulation (material flow and DMU) which has a strong process-oriented approach in regard to the real manufacturing systems.

The main PhD thesis objectives are: development of the concepts needed to improve data migration from the simulation to the real manufacturing systems (also vice-versa), identification of current and future technologies that may incorporate the concepts' aspects but also the validation of the proposed concepts by a first prototype, that uses current available technologies, for a better interconnection of the manufacturing simulation with the manufacturing systems' control units.

The proposed concepts fit the current research trends of the SOA paradigm applied for automation, and, the proposed architectural approach augments today's research fields with additional aspects, imposed by a more precise construction of the manufacturing simulation and an improved migration between simulation and manufacturing systems.

The architectural approach, new from the simulation and manufacturing standpoint, is based on the splitting of the whole manufacturing process intro "atomic" sub-processes (depending on the technical limitations and manufacturing specific characteristics) but also the embedding of the whole relevant aspects for simulation and the automated manufacturing systems into services (McS- Manufacturing-centered Services and Acs-Automation-centered Services).

Although the first international steps were made, at the industrial level, concerning the service orchestration for controlling an entire manufacturing system (SIRENA approach), yet because of the lack of augmentation of the simulation relevant aspects within a service and the lack of concepts demonstration by means of prototypes, the applicability is still low compared with the potential results.

Thus, a systematic research, concerning the augmentation of the current SOA approaches, with the simulation relevant aspects, and the proposal of a new simulation architecture to obtain an improved agility, is worthy of research, due to the large application spectrum concerning the industrial implementation.

1.2 Thesis evolution and structure

The PhD thesis topic inscribes itself not only into the international Digital Factory preoccupations but also in the ones found at "Lucian Blaga" University of Sibiu and at the University of Kaiserslautern - Germany with their mechanical and automation departments though their innovation efforts implemented in the SmartFactory^{KL} research platform.

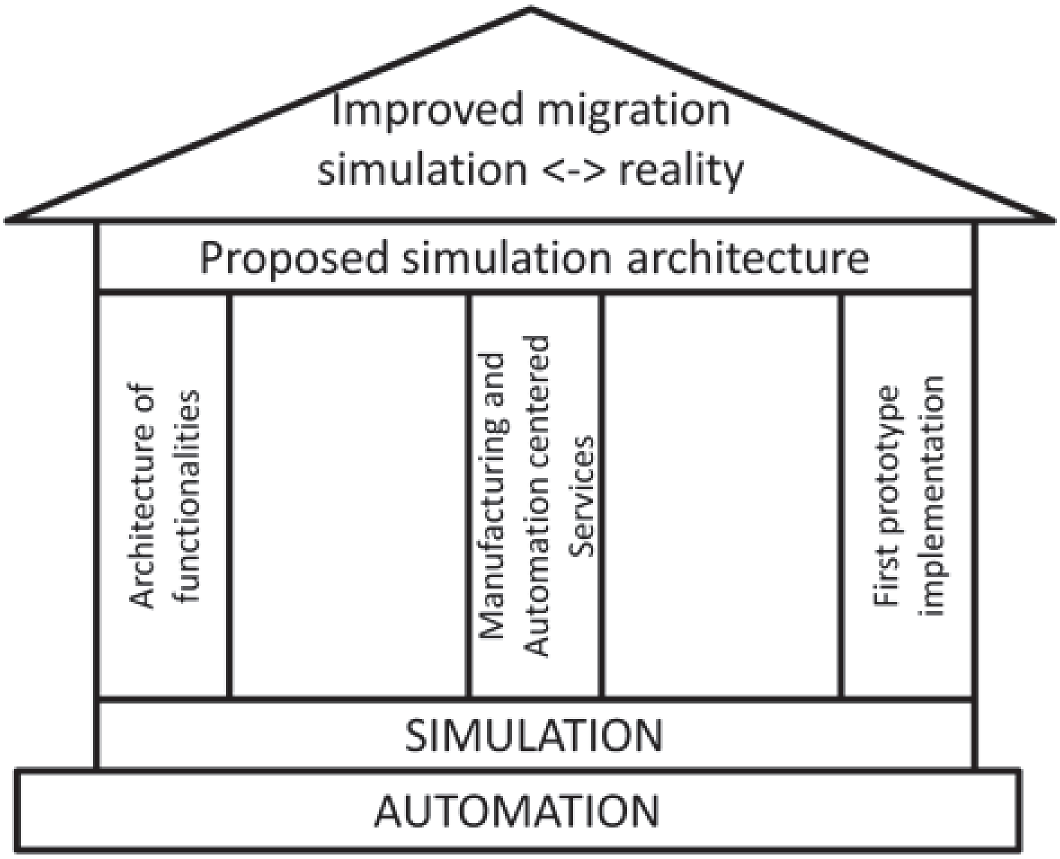


Fig. 1.1 PhD thesis evolution

The PhD thesis formulation was based on the conclusions formulated after the state of the art was researched and analyzed, concerning the automation field, manufacturing systems modeling and simulation, factory evolution and also the relevant IT technologies for manufacturing – mainly the SOA concept and its implementation technologies.

The thesis has evolved (fig. 1.1) with the analysis of the automation state of the art but also with the manufacturing simulation approaches. The issues formulated after the analysis of the state of the art can be solved by the approaches represented by the McS and AcS, the methodology to determine the “atomic” sub-processes. The concepts were validated though a first prototype implementation. Thus, the main objective of improved migration between manufacturing simulation and the real manufacturing system is achieved, which also contributes to a higher degree of company agility.

The PhD thesis is structured into seven chapters, and the first chapter focuses introductive elements and also the motivation and the thesis objectives.

The second chapter summarizes the state of the art concerning the automation field, manufacturing systems, Digital Factory concept, manufacturing systems modeling and also the SOA concept and its current implementation technologies. Concerning the automation perspective, the current and future technical solutions are presented regarding the main automation actors: programmable logical controllers (PLC), computerized numerical controls (CNC), motion controls and also communication networks and middleware approaches. Regarding manufacturing systems, the second chapter focuses also the factory and traditional and reconfigurable manufacturing systems evolution. The Digital Factory concept is presented from the technical literature standpoint and it is defined in the current PhD thesis context. Manufacturing systems modeling techniques are too summarized in this chapter, with graphical, mathematical and simulation methods. Besides the SOA concept, which is present with both meanings for IT and automation, the technological preferred implementation solutions of this concept are likewise shortly presented.

In the third chapter the problem is thoroughly presented from the state of the art analysis and synthesis and also the PhD thesis objectives and chosen solving approaches are formulated.

In the forth chapter the concepts which allow the solution of the formulated problem are presented: the architecture of functionalities in the context of the McS and AcS services. Besides these, a methodology is proposed to split (mesh) the entire manufacturing process into -„atomic” sub-processes and also the technologies which may enable the development of the McS and AcS.

In the fifth chapter the first prototype implementation of the proposed simulation architecture is presented using current manufacturing simulation software and control units.

At the end of the thesis, with the sixth and seventh chapter the conclusions are synthetically presented from all of the PhD thesis phases and also the future research directions are formulated.

Research made within the “Lucian Blaga” University of Sibiu doctoral school, within the POSDRU/6/1.5/S/26/7706 project, named „Creşterea rolului studiilor doctorale şi a competitivităţii doctoranzilor într-o Europă unită”, co-financed by the European Social Found through the Operational Programme Human Resources Development 2007-2013.

2. State of the art

2.1.1.1 IEC 61499 – open standard for distributed control and automation

The IEC 61131 standard, which has been widely adopted by the industrial automation industry, is probably reaching the end of his technological life cycle, mainly because it is not able to meet the demands for "agile" manufacturing (automation systems which are distributed, decentralized and flexible regarding the ease of reconfiguration) [98]. The IEC 61499 function block standard, was first standardized in 2005, and plans to solve the later presented issues [96] that the IEC 61131 is faced with.

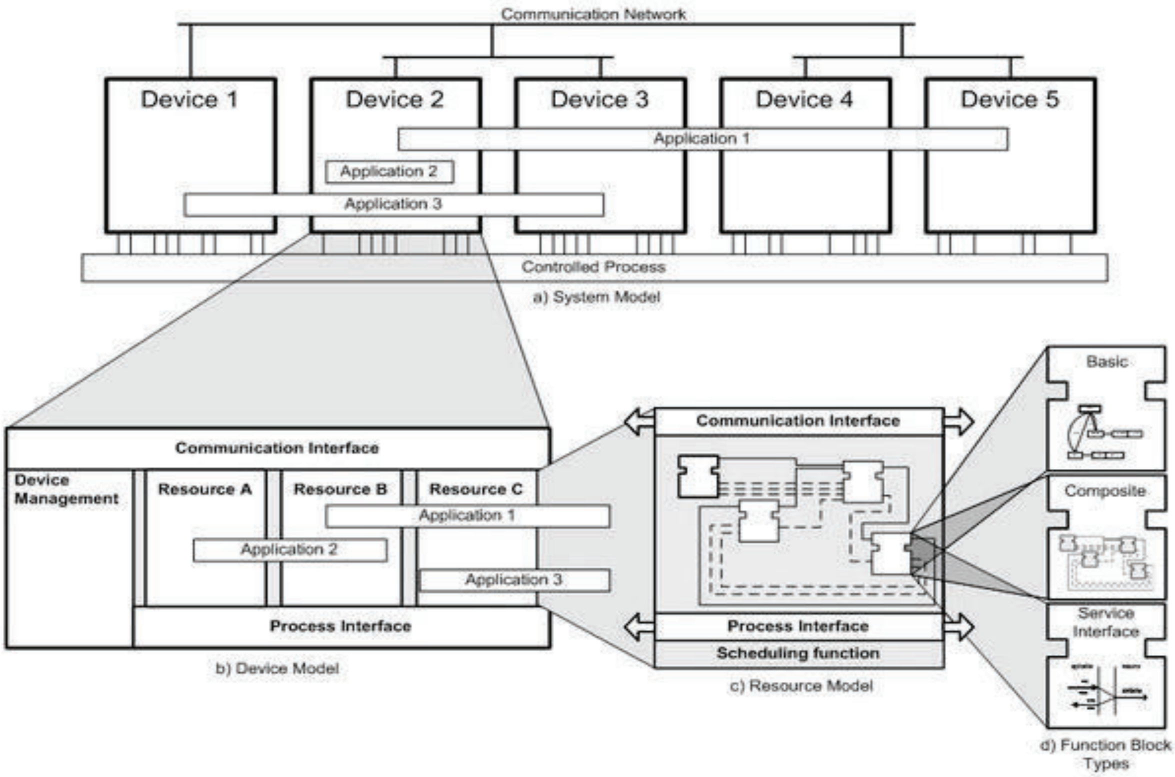


Fig. 2.1 IEC 61499- Reference Models for Distributed Automation [120]

The IEC 61499-4 standard's mission strives to the following attributes of systems, devices and software tools [96] [127]:

- Interoperability: the ability of devices from different manufacturers to operate together and to perform the functions specified by one or more distributed applications;
- Portability: the ability of software tools to accept and correctly interpret library elements produced by other software tools (other manufacturers).
- Configurability: the ability of devices (from different manufacturers) and their software components to be configured (selected, assigned locations, interconnected and parameterized) by multiple software tools (designed by different manufacturers).

Features regarding the ability to support dynamic reconfiguration were also included in the IEC 61499 standard [96].

The “building brick” of the whole IEC 61499 standard is the function block (FB). The second is the usage of an event-based execution model.

2.1.2 Traditional and reconfigurable manufacturing systems

A reconfigurable manufacturing system (RMS) is a system designed from the beginning to allow a rapid change of its structure, its hardware and software components, in order to easily adapt the production capacity and functionality within a specific parts family [17].

RMS represents the advantages of both „traditional” manufacturing solutions, by adopting from dedicated manufacturing systems the focus on specific family of parts, and from flexible manufacturing systems, the possibility to adjust in case of a production variation without changing the production scheme.

Reconfigurable hardware and software are necessary but not sufficient for a RMS. The core or the reconfigurable manufacturing systems paradigm is the approach based on system and open reconfigurable controller design implemented intro modular machines that may be constructed by synthesis of the moving modules [45]. A RMS system should assure reconfigurability of: the whole system, the machines hardware and control software system.

Such a RMS system may be easily compared with the „Plug & Play” system from the PC industry that offers an easy configuration possibility corresponding to different needs of the PC system.

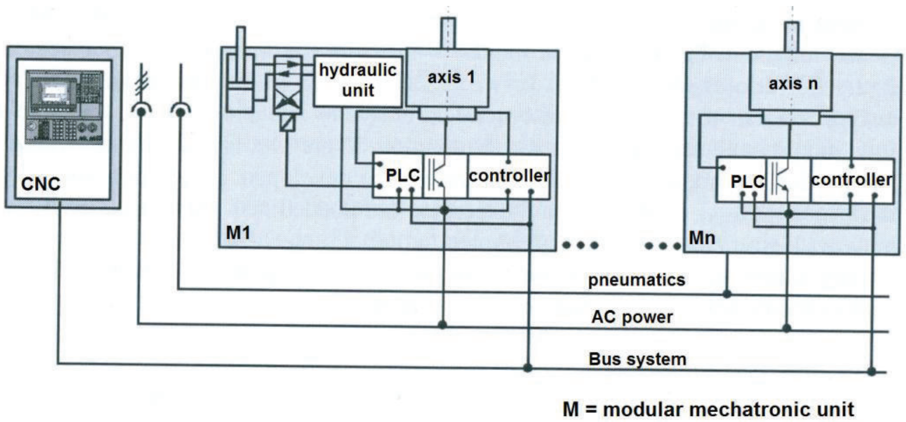


Fig. 2.31 Modular mechatronic units that compose a RMS[83]

The entities composing such a RMS system should be composed by easily connectable mecatronic units into different manufacturing configurations. Such an example is depicted in 2.31, in which these mechatronic units are composed out of: a PLC, a drive controller, a convertor, rectifier and a hydraulic unit [83]. Such a system will allow the interconnection of various modules in a couple of minutes.

The vision of such an adaptable autonomous system is depicted into figure 2.32.

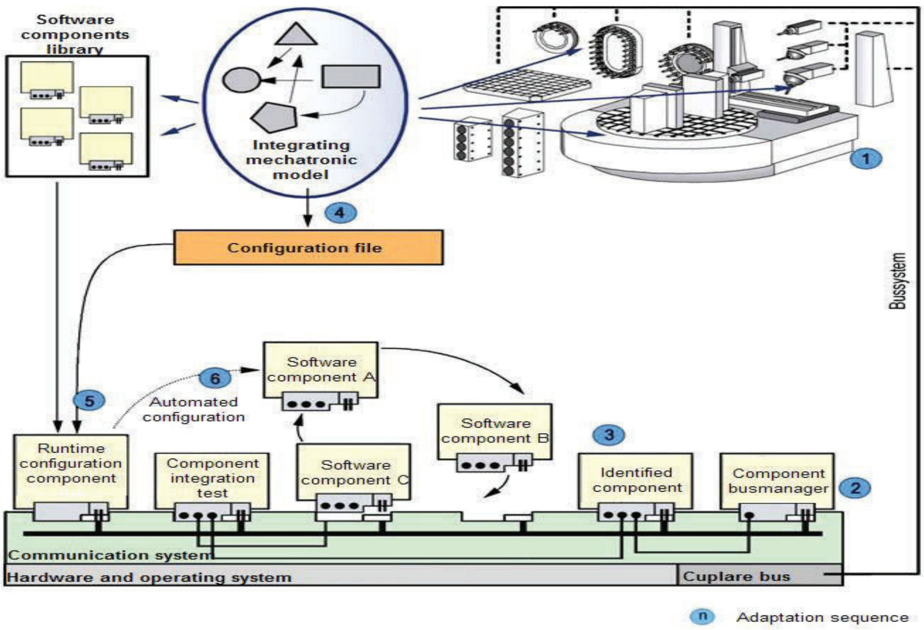


Fig. 2.22 Vision of an adaptable and autonomous control unit [84]

2.2 Digital Factory

2.2.1 Introduction

The current context can be characterized by complexity enhancements required by customers, global competition and a very fluctuant customer demand regarding the product volume and product type. Remarkable product diversity can be observed to cope with the global client requirements. A good example is the auto industry, where the major manufacturers have tremendously developed their product range compared with the 1990s to cope with the global demands (fig. 2.33)

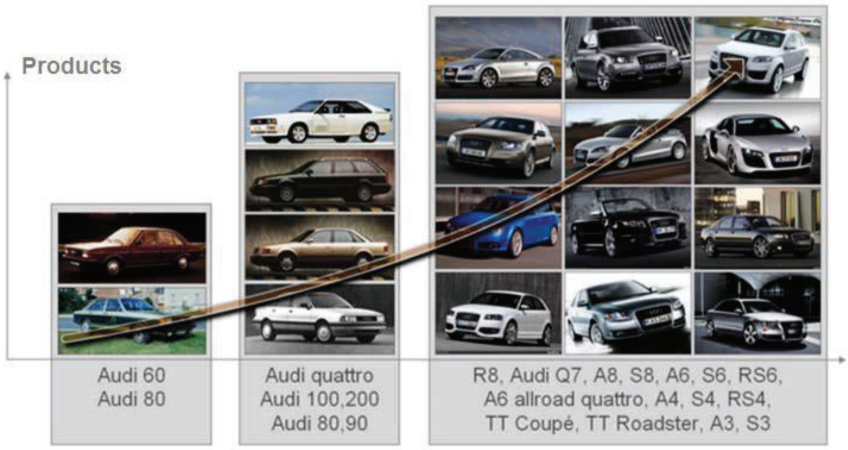


Fig. 2.33 Audi product diversity evolution [94]

At this point there can be distinguished a separation between the engineering tools and manufacturing systems. There is no cohesion between product and process development and simulation with the production. Even the digital tools are not yet fully integrated, thus, the so called digital factory engineering tools are still far from integrating the two worlds (fig. 2.34).

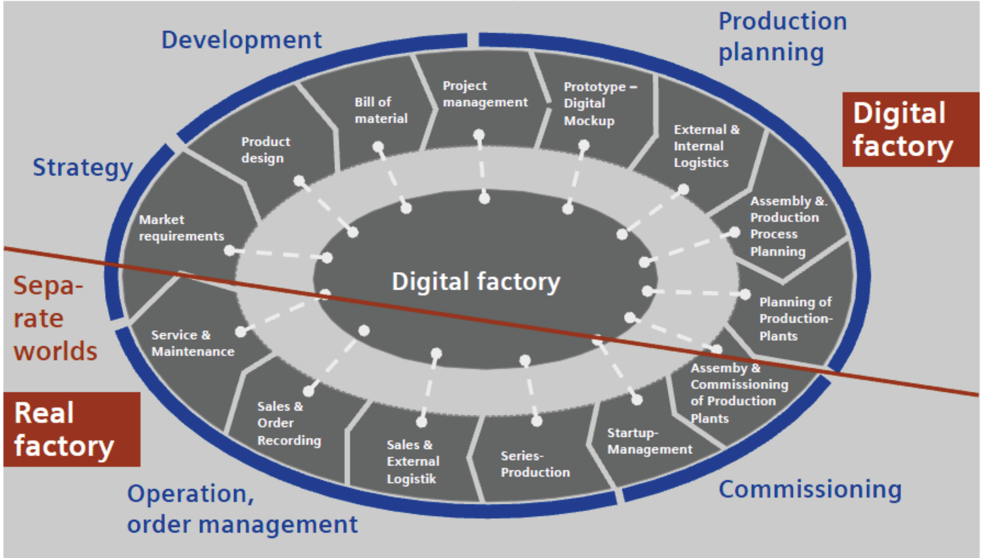


Fig. 2.34 Separate worlds (real factory and Digital Factory) [89]

2.2.2 Definitions

The Digital Factory concept will be used within this thesis in conformity with [112], having the following definition:

The Digital Factory represents all necessary conditions in order to create an efficient virtual and real manufacturing respectively the efficient factory.

2.2.3 Manufacturing simulation

Simulation is the behavior imitation technique of a system or process (e.g. mathematical, mechanical, social etc.) by means of an analogue artificial corresponding system (or process). Simulation can be seen as a way to observe the outputs of a model depending on its various inputs [9] or it can be seen as an imitation of the process operations during time [4]. In other words, simulation can be identified as a behavior analysis method of a real system that also can determine the effects of exogenous interventions regarding this system.

Simulation is used largely in the industry to model, study and optimize existing or future processes. Discrete event simulation (DES) is a branch of the simulation methodology largely used in industry to analyze manufacturing systems. DES is based on stochastic discrete models.

2.3 SOA – Service-oriented Architectures

2.3.1 Introduction to SOA

The “service orientation concept” is regarded as “an ideal vision of the world in which resources are cleanly partitioned and consistently represented” [22]. The basic idea of service-oriented architectures is that all involved software modules – which are called services – are self describing and can be integrated in an interoperable way [38]. The self descriptiveness bases on a model based company wide description of communication and information models. Reconciling these two principles (building an autonomous system that is yet interoperable) is very difficult.

SOA is and was used mainly in IT and represents software modules which are executing a special functionality within a business process.

SOA is neither a technology nor a technology standard. It represents a technology independent, high-level concept that provides architectural blueprints. At this point the preferred implementation mean of SOA is through Web Services. Web Service technologies (e.g. UPnP, DPWS, Web Services) are communication methods between electronic components over a network (Internet) using technologies like: XML, SOAP, WSDL, WS-Addressing, WS-Policy, WS-MetadataExchange and WS-Security [37].

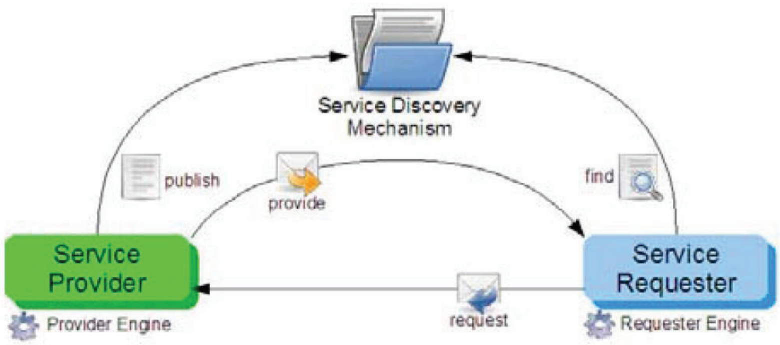


Fig. 2.46 Service communication mechanism within the SOA concept [57]

An important characteristic of SOA is the loose coupling of services. The term coupling refers to the degree to which services depend on each other. An agile service environment needs services that are independent to each other. Thus, service compositions can be done quickly and with high flexibility. In normal software architectures loose coupling can mainly be obtained by software modularization and a high degree of cohesion.

The service access is described through its interface. Thus, the requester will be able to access the service without focusing on the service implementation. In order for services to interact with each other they shall be first discovered. This is done by means of a service discovery mechanism (fig. 2.46). So the SOA mechanism has three components: service provider, service requestor and a service discovery mechanism. Messages are the elements which are connecting the three components

2.3.2 SOA in automation

For SOA applied to the IT the blueprints are focusing on the slicing and composition of enterprise applications as services which are technically independent and have a direct relationship to business functionality. Thus, a service is a software component that executes a special functionality within a business process. In the automation domain meanwhile, a service encapsulates mechatronic functionality. The main difference between these two application fields is the dependence on the location of the hardware that executes the service. While it does not matter where the physical execution of a piece of software takes place, the right execution of a mechatronic function that impact a physical process is heavily dependent on the location of the hardware. A detailed comparison between SOA in IT and automation can be found in [66].

Loose coupling of technical processes additionally requires modular design of the mechatronic components in order to rearrange the hardware structure easily. Thus, the hardware of the automation devices and the control programs have to satisfy the paradigm of loose coupling. The ideal case would be SOA applied till the lowest levels, main objective of the SIRENA perspective [145]. An agile service environment needs services that are independent to each other. Thus, agile manufacturing systems need loose coupling both at the hardware and software level.

The benefits of the SOA approach to automation could be summarized as follows:

- Connection of different automation components could be done with ease without current integration issues;
- A process could be implemented by a certain service succession invocation;
- Programming doesn’t deal anymore with I/Os and the logic is correlated to the functional perspective of real manufacturing processes. We are discussing about a higher level of abstractization;
- A high level of independence could be achieved concerning the hardware and software and device reutilization;
- The automation pyramid becomes hetearchical.

2.3.3 Orchestration versus choreography

There are two approaches for the execution of manufacturing systems having a SOA approach: orchestration-based or choreography-based.

The orchestration logic is concentrated in one place and can therefore easily be changed should the machine need to be extended or reconfigured. Furthermore, the absence of direct communication between the individual devices greatly facilitates the machine set-up process as well as the replacement of devices. On the other hand, more time is spent in exchanging messages between the devices and the orchestrator than in case of direct device-to-device communication. This may or may not be an issue, depending on the dynamics of the production machine

While orchestration is an approach oriented towards the execution of processes or succession of atomic processes, it does not take into consideration different communication patterns needed to invoke associated services with these atomic processes. Choreography, one the other hand, defines rules that define messages and interaction succession that need to take place in order to execute a specific process through a particular service interface [39].

3. Problem formulation, thesis objectives and approach

3.1 Current context

The global market imposes with it a new dimension: the global competition. Companies are faced with constant customer requests that are generally represented by: quality products (highly customizable) that are fast available (generally speaking a car is outdated in three years and a phone even in a couple of months) with a low price tag that arrive on time with the right specifications at the right place. In this context the companies who will survive and strive will not be the strongest or the smartest, but the ones better adaptable to change.

Different existing concepts intend to guide the path towards achieving fully "adaptable" companies at this point. One recent concept, dealing with the issues stated above, is the Digital Factory. The concept states that in order to achieve attributes like adaptability, there should be a strong link between the real factory and a digital counterpart of this factory. The vision is to optimize all parameters in the virtual world (proactive or during operation) and then to send the data to be used by the real production line. There should be a strong link between these two entities. The concept is thoroughly presented in the VDI 4499 standard [151].

One of the major issues deriving from this ideal is the communication between simulation and the real production line, stated in a general form, or, more specific, the migration of an optimum production scenario (done obviously by virtual models) to a reconfigurable production facility. A complete and error-proof migration between the two actors should offer the agility needed to survive and strive in the global market environment. Communication between simulation and the real line requires in the majority of cases the usage of OPC (Ole for Process Control), since only messages can be exchanged.

Major issues stem on one side because of the automated line - with its various controllers (PLCs stem from different vendors) - and on the other side because of the software tools both of which, from the beginning of development, were never intended to provide functionality or mechanisms for "transferring" results from one to the other – simply said they are lacking cohesion. Each perspective (automation and simulation) over the same manufacturing processes is completely different and not complete (one deals with low level programming and the other mechanical aspects – kinematics, collisions, part design. This is one neuralgic point that does not allow an easy and smooth migration from one environment to the other.

At this point there is only a state change information exchange (triggering I/Os), data about processes are never exchanged. When a problem will occur at the PLC programming level in the real factory (likely when considering that no simulation is and probably on midterm will not be 100% accurate compared with the real line) today's limitations will still apply (someone will have to understand the PLC programme, interpret it and correct it). No current approach manufacturing approach is real-process centred.

3.2 Current simulation approaches

In most cases, manufacturing simulation software (material flow, Digital Mock-Up) has reached this level by adding different functionality or tools during its evolution. What remains the same is the basic logic or the architecture that was first conceived. Simulation software tools were designed to give engineers a better understanding of the system and take according actions. The needed actions were then presented to the execution level where the automation and production engineers tried to do their best of implementing what they have understood. This is the past and present way of doing things.

One current possibility is to model the behaviour of the system by internal logic within the simulation. By this, the various states of the system (relevant for the controller) are modelled by Sequential Function Charts (SFC) in Delmia [88] and global variables in Plant Simulation [80]. Then the SFCs or the global variables are then linked with I/Os (ports). When a certain event in simulation happens their state (SFC or the global variable) also changes (this will trigger an event –via OPC - at the controller level).

The other option (in case of Delmia) would be to programme the logic (PLC programme) on the PLC and to connect I/Os from PLC's to the simulation ports (visualisation of the logic in an animated virtual world). Obviously a mix of those two is also possible.

Leitao et al. underlines in his paper [49] that Delmia Automation (other tools offer even more limitations in terms of validating and migration process from virtual environment to the real world) enables validation of the control logic in a simulation environment and offers the possibility of programming various PLCs. But, the migration to the real line has the following obstacles [5]: validation is achieved by means of simulation model and "a validation using formal language is missing"; deployment into controllers is dependent on developed interfaces, "requiring more integration and interoperability compatibilities"; it is not prepared to allow "multi-agent systems and service-oriented ecosystems".

To conclude, even probably the best digital manufacturing tool today (Delmia) doesn't offer a "smooth" transition from the virtual environment to the real world and vice versa.

3.3 Current PLC approaches

The IEC 61131 standard has been largely adopted by the automation industry, but could be replaced by the IEC 61499 standard – open standard for distributed control and automation – that promises to bring the agility needed to face current and future exigencies and offer attributes like: interoperability, portability configurability and dynamic reconfiguration.

The IEC 61499 standard is based on function blocks and an event-based execution model. It allows for an application to run over several resources of different devices, thus offering true distributed controlling capabilities.

Worth mentioning is the ISaGRAF environment, which is the first commercial available tool to support the Function Block (FB) standard (IEC 61499) and also the IEC 61131 standard.

3.4 Current middleware approaches

The first main wide-spread middleware solutions were OMG'S CORBA (Common Object Request Broker Architecture) and the (D)COM (Distributed Component Object Model) architecture from Microsoft. The common goal behind CORBA and (D)COM was to provide a technology to allow various applications to communicate over a network. Although CORBA was developed first (around 1991), it has at its roots some of today's concepts and trends, with its SOA fitting and platform independence, differentiating itself from the "younger" (D)COM approach. The main advantage of Microsoft's (D)COM technology was that everything was already implemented and available with Microsoft Windows operating system. This is the main reason why the Microsoft technology became so spread.

OPC UA (Unified Architecture) keeps all the functionality of Classic OPC but uses state of the art Web services technology and an optimized binary TCP protocol for high performance communication instead of Microsoft's COM/DCOM technology. Thus OPC UA becomes platform-independent overcoming one of the major issues of classical OPC. OPC UA specifies how data is exchanged, while standard information models specify what information is exchanged [56].

OPC UA offers the possibility to use information models (some basic ones are defined). Using already defined information models one can define other information models which are custom. With the introduction of metadata it is possible not just the transmission of variables, but at the same time how the transmitted data should be interpreted (data over data).

The major downside at this point is the data transmission speed with OPC UA. Because more information is exchanged the data transmission speed is low (compared at just a couple of variables, it is clearly slower than classical OPC) [56]. Another down side would be that not all aspects of OPC UA are yet available, like for example the server to server communication.

3.5 Problem formulation

An approach that links the manufacturing simulation (process flow and digital mock-ups) with controllers while having a holistic view of the real processes that take place in manufacturing is missing. Such an approach can become future industry implementable, when considering today and future controller and middleware advancements. They are shortly described in the following paragraphs.

The IEC 61499 open standard for distributed control and automation can be considered suitable in the above stated approach because of the application model described in the standard (e.g. a pick and place service), which is able to run over multiple devices (it could involve one or more robots, a conveyer, sensors and various actors etc.), thus allowing for a true process orientation approach. This way we can distinguish the real processes that are happening in the factory and programme the controller accordingly (by having the overview of the process) and not focus on I/Os.

The application model from the IEC 61499 standard and its integration within the controller could be also extended to the simulation models that have service orientation. Simulation should provide a better outlook of the process that take place in the factory by having this view. Thus the same perspective of the same real process will exist in both cases (of the simulation engineer and of the automation engineer).

As middleware, OPC UA is capable of exchanging more data between the automation pyramid actors compared to its predecessor. It is suitable for our before stated approach because: it is service oriented, capable of exchanging more data, with a better security, platform independent and continuously updated and implemented (or future implemented) by major industrial companies like Siemens, Rockwell Automation etc.

A process (service) oriented manufacturing architecture involving simulation and controllers is the missing link between a better coupling of the digital world to the real world. As presented above the advancements in controller technologies and middleware technologies will allow for a process oriented approach from simulation to the real production lines.

3.6 PhD thesis objective

Goal of this thesis is to propose a novel architecture in case of manufacturing simulation (material flow and digital mock-ups) that has a strong process orientation approach regarding the real manufacturing systems. Thus, the thesis accent will be directed mainly towards the concept level.

By simulation term, in the thesis context from now on, it will be understood material flow and DMU (Digital Mock-UP)

3.7 Approach strategy for the PhD thesis objectives

To reach the thesis goals the followings concepts are proposed: the architecture of functionalities and the manufacturing-centred and automation-centred services (McS and AcS). The architecture of functionalities presumes the splitting of the entire manufacturing process – considering process specific characteristics and mechatronic aspects- into „atomic" processes. The splitting process should primarily focus a high degree of correspondence between the „atomic" process and its corresponding executing mechatronic device/devices. Thus, the whole manufacturing process could be shaped as a succession of the "atomic" processes (fig. 3.3).

In order to allow the atomic processes description and a better information transfer between manufacturing simulation and manufacturing systems the services - McS and AcS - will be developed as concepts. These, are developed in order to incorporate the basic ideas of the IEC 61499 standard, current SOA approaches in automation but also current and future technologies that allow the storage and transfer of necessary data to fully represent the manufacturing processes (at the controller level and simulation level) using a single neutral data exchange format (e.g. AutomationML).

The proposed simulation architecture will incorporate the ideas from the architecture of functionalities and of the Mc Sand AcS - services. Its applicability and testing will be presented into the fifth chapter, as a proof of concept, that will use current market available technologies.

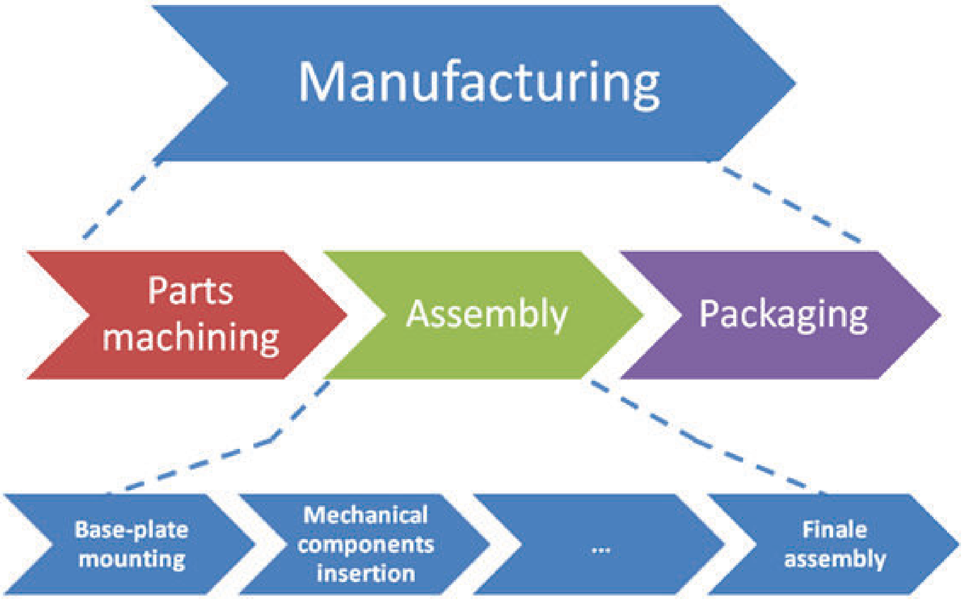


Fig. 3.3 Architecture of functionalities example

4 Proposed concepts

This chapter deals with the motivation and description of the service-centred concepts behind the Manufacturing-centred and Automation-centred Services that will enable a better linkage of manufacturing simulation with the real manufacturing systems. This is done by encapsulation all the relevant manufacturing aspects (for simulation and control unit) under one entity, represented by the generic service.

4.1 Introduction

The execution of a SOA-centred manufacturing system implies the usage of service orchestration or choreography. Thus, regarding the orchestration approach, the whole manufacturing process execution at the logical execution level, concerns only the succession of service invocations by the orchestrator.

SOA implies not only the creation of independent and interoperable logical units – the services - but also a modular (independent and interoperable) mechatronic construction for a improved hardware reconfigurability and not only logical reconfigurability.

SOA at the hardware level will most likely imply, besides standardization and mechatronic modularity, the introduction of computing capabilities at the lowest levels within the automation pyramid.

To support a SOA manufacturing scenario, considering the most general case, modern manufacturing systems have one or all of the three major components:

- Smart mechatronic components: that have built-in control and intelligence
- Aggregated mechatronic components: built from various number of basic mechatronic components governed by an internal process control component (control and intelligence). A good example in this case would be a transport system.
- Upper process control component that governs over the whole manufacturing system.

Purely mechanical or electromechanical components could be linked with these three major components. But we will focus on systems without these components which have or are linked with control and intelligence.

Imagining a complex production system having each sensor/actor or simple device exposed as an service it is easily recognizable that the orchestrator has to an overwhelming level of information in a short time stamp and this will lead to major lags or communication breakdowns (because all state changes should be interpreted by only one orchestrator). To avoid this services should be grouped into larger services - compound services - with their corresponding orchestrator. This way the “master orchestrator” has to deal with only a few “compound services” in case of production and this will not affect the overall data transfer speed. It makes sense to group all services from a device into a couple of compound services (also taking into consideration the linkage with other exterior devices) and a device orchestrator for executing the compound servicess. The “Main Orchestrator” orchestrates exposed compound services and he also manages service synchronization when a process involves at least two services (compound) over two different devices.

A fully SOA implementation at the manufacturing systems’ level is not possible mainly because of the following causes:

- Technological: development of new or derived devices from the existing technologies for manufacturing (automation) to host services until the device level;
- Software: additional aspects that need to be embedded into services should to be defined beside standard defined by Web standards;
- Methodological: how the big real manufacturing process can be split up into well-defined technical sub-process (e.g. services) that are autonomous, interoperable and reusable.

4.2 Manufacturing and automation-centred Services (McS and AcS)

4.2.1 Premises

A service-based manufacturing and simulation architecture represents a complex hardware & software lego-system designed to easily build, commission, produce and reconfigure the manufacturing system in order to meet the market needs. Beside the fore mentioned prerequisites, the integration with higher enterprise levels (ERP) should be kept in mind. Identification of the main “lego” constructs regarding different phases of the manufacturing system will lead to the definition of the main services that can be then used into larger, compound services. In our approach the services are orchestrated and not choreographed.

For an easy *system construction,* services should enable:

- Discovery, identification and integration of the newly inserted device into the system;
- Location of the physical device within the system;
- Description of the device;
- Functionality of the device.

For a *thorough commissioning*, services should enable:

- Verification of services regarding deadlocks, stability, existence of unforeseen prohibited states;
- Full mapping of logical and physical characteristics between real line and simulation:
  - ✓ Migration of physical and functional characteristics data over a local network or internet to the simulation environment and vice versa from the simulation environment to the real line to enable a full simulation description or a full functioning real manufacturing system;
  - ✓ Process logic succession migration from the orchestrator to the simulation and vice versa;
  - ✓ Hardware in the loop (HIL), software in the loop (SIL) or both testing scenarios;

For a smooth production, services should enable:

- "Fast" or even hard real time updates of states, variables, emergencies etc.;
- (Pre-)Visualization of previous, current and future processes taking place in the real manufacturing system on the real line and in simulation;

For a complete reconfiguration, services should enable:

- Update of actual hardware and logical characteristics of the line (deletion of old ones and insertion of new ones);
- All aspects from the commissioning prerequisites.

For migration of data to higher enterprise levels, services should enable:

- Non real time update of manufacturing systems values or characteristics (parts produced, failures, physical devices involved in current production, etc.)
- Non real time update of physical system devices in higher enterprise level systems after a reconfiguration action.

To summarize there are three main scenarios:

a) Only the manufacturing system is working: stopped, stand-by, work in progress. Information into the simulation or higher enterprise levels should be updated when requested.
b) Simulation and real manufacturing connected: SIL, HIL, upload and download of data between the two actuators. At the end the real manufacturing lines is updated with the latest process execution order.
c) Simulation only, when new configurations are pre-tested. Inputs from manufacturing line are necessary to upload the devices physical characteristics and the process logic.

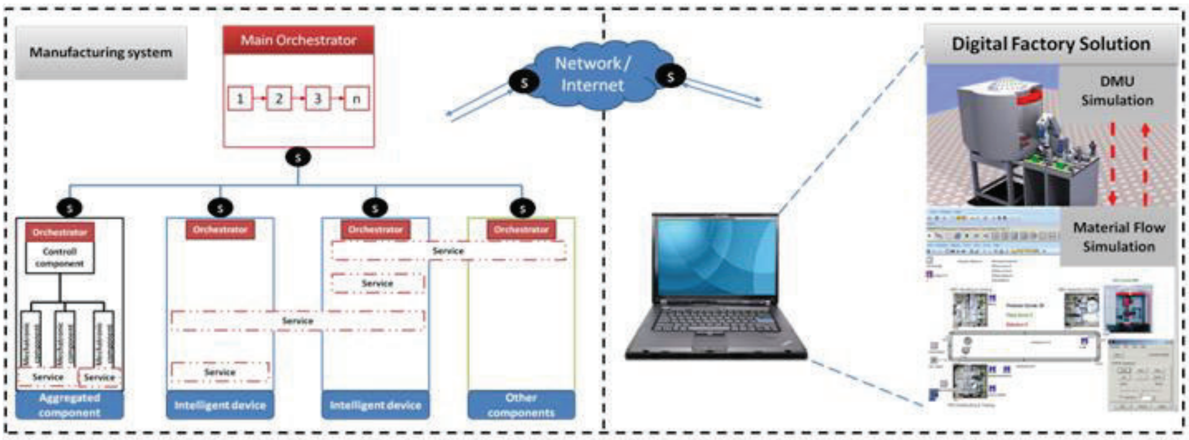


Fig. 4.2 Example of a simple service-based manufacturing & simulation architecture

According to these tree main situations two different main service specifications underline themselves:

- One "light" aspects' structure for manufacturing only situations (update state, variable etc. for higher enterprise levels or visualization possibilities on the line and in simulation). These "light" services will be named Automation-Centered Services.
- One "heavy" aspects' structure that comprises the "light" one and beside this offers the capabilities for migration of data from real line into simulation (and vice-versa) and thus enable commissioning and reconfiguration actions. These compound and complex services will be named Manufacturing-centered Services.

These two AcSs and McSs highlight different aspects of the same real process, but depending on possible scenarios (the three described before) certain aspects from the most general service are being "hidden or used" to form the AcS's light perspective, or, on the other hand all the aspects of the general service are being shown to form McSs. AcSs should embed just enough aspects (fully or partial) that allows a line to function on its own and update data for insertion into higher enterprise level reports or databases.

Why there should be two perspectives - AcSs and McSs? There are three major reasons:

- AcSs may be hosted by future sensor an actuator elements with reduced computational capabilities (SIRENA perspective [145]). McSs will comprise the whole functionality in the main computational unit (main controller). SOA could then be implemented also at the hardware level.
- Security. All AcSs ensure the manufacturing system is functioning. Full description is available only in commissioning or reconfiguration scenarios. Complete data is not accessible for unauthorized persons.
- Automation. The automation engineer can deal with AcSs for simple tasks or simple process changes within the orchestrator.

4.2.2 Generic service concept definition

Services in general, are software modules execute a special functionality within a business process. This is fine in case of a business process but not enough in case of a manufacturing system. Very important and similar with an application define within the IEC 61499 standard (distributed control and automation), in this presented approach, a service should be able run from a single resource on a single device to various resources of different devices (even all resources of all devices in the most extreme case). In our approach we view a component as a device. A mechatronic device is composed out of mechanical, electrical and computational elements. These elements should also be visible for someone who is using a specific service not necessarily used when the real production line is working but if it is the case of building a system from ground up. Thus, some other characteristics a service should comprise, besides the general ones considering current Web-standards specification (identification of the service, the control logic and the component's functionality), in order to adopt it for the complex manufacturing systems. These could be:

- Description of the device's physical characteristics (mechanical, electrical, computational) - devices over which the service runs. Very important in case of malfunction or in the case of the manufacturing system connection with simulation. Physical characteristic could be imported directly in the simulation thus achieving a correct physical description.
- Physical relationship of the device to the manufacturing environment. Where is device located within a manufacturing system related to other devices?

- An implementation characteristic. This should link together the functionality of the service (read, write in case of a RFID device) with the physical description via a communication technology. This deals with the internal inter-connections within a service regarding the hardware upon which it is executing on. Traceability of physical logic with the service functionality can be this way achieved.
- Aggregation characteristics. Our service should be viewed as a holon. It can represent itself a whole but also a part of another whole (sub-whole). Thus, inter-connection with other services (at the same lever or regarding upper levels) should be foreseen at its interface capabilities to create a "compound" service. This will allow building complex services to fully describe the manufacturing of a specific product.
- Service orchestration or service choreography characteristics regarding the service. The selected approach will definitely impose different prerequisites from the architectural standpoint of the service. The later implies surely an additional interaction because there will be no more an orchestration from a fixed point but rather an negotiation between various services in order to achieve a service process model (which in case of the service orchestration resides in the orchestrator).

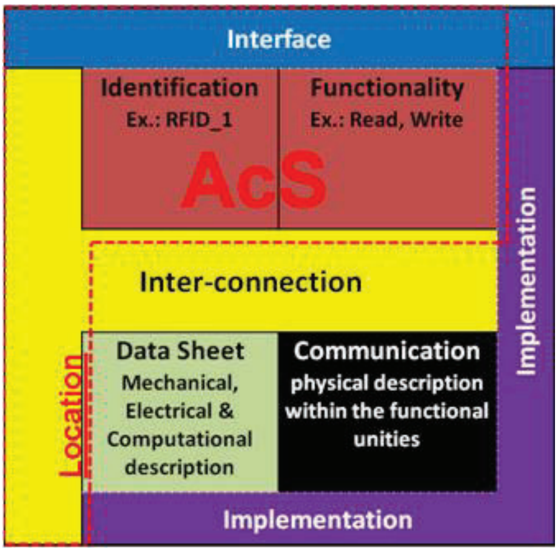


Fig. 4.3 Aspects encapsulated by AcS

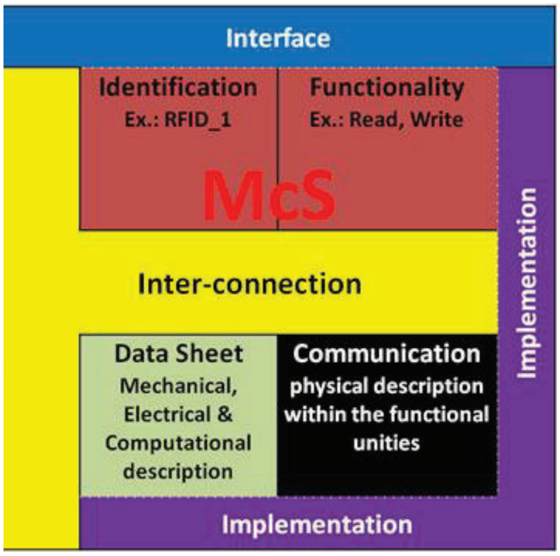


Fig. 4.4 Main aspects of a general service – the McS

Considering the previous stated characteristics for services regarding production manufacturing we may define such a service as *an context independent software application compound that encapsulates and interconnects the identification of the service* (basically its unique name), *functionality* (e.g RFID_1: read, write), *physical description of the device and its relationship to the environment* (data sheet) *with the physical communication technology*. We will name this complex software compound which is this most general service for manufacturing (later named as Manufacturing-center Service – McS – fig. 4.4). Thus a general manufacturing service has these main aspects:

1. *Identification*: it is necessary for an unambiguous service selection or addressing;

2. *Functionality*: it is necessary to effectively identify which and what functionalities does the service have. The functionality aspect is composed out of sub-operations their corresponding parameters and description (a RFID service will have read and write operation- see table 5.3 from [65]). Nevertheless the computational unit that is running the functionalities has to be also regarded.
3. *Data Sheet*: within the service this aspect serves for the mecatronic description and location of the stand alone device or devices on which the service is running. The device's relationship to the production environment should also be described (e.g. location of the device). It has these main elements: mechanical, electrical, pneumatic and logical. This is a storage file of all the needed data and interconnections to build current or previous virtual representations of the system into the simulation environment. Necessary for building a system from ground up (in simulation or reality).
4. *Communication*: this aspect reveals which physical communication linkages exist regarding the physical environment and which communication technologies and maps are used for service to service communication. In the most physical general case this would involve machine to machine communication over a network, between devices or between resources of within a device. Necessary for diagnostic purposes.
5. *Implementation*: this aspect comprises all the needed elements that allows for a mecatronical unit to function stand-alone. Many, if not most, of the needed aspects are the ones encapsulated into the functionality and the communication aspect.
6. *Inter*-connection: a very complex aspect of the service. It is like "binding material" that allows the aggregation of the previous aspects into one unity and also for the aggregation of basic services into composed ones (that are running from one device to a finite number of devices). This could be represented by the *Enterprise Service Bus* when thinking of the most complex compound service at the highest level. (e.g. Produce the X product). Interconnections could be structured into two parts: internal and external inter-connections. Internal inter-connections are complementing the implementation aspect in order to allow an update of relevant "components" of the Data Sheet. This will allow, when needed, an update of the current production phase into various components of the data sheet that can be shortly uploaded into the simulation system or higher enterprise level systems. The external inter-connections are responsible for coupling services to each other to form a higher compound service. This will also have to deal to the succession of each sub-service that it is made of.
7. *Interface*: is used to access/discover the service from outside the service.

Figure 4.5 identifies the general manufacturing service's aspects within more devices (for representation simplification the device uses also a SOA approach to its design). A device has from 1 to n resources. One resource groups together mechanical electrical and computational elements that in the end offer a specific functionality.

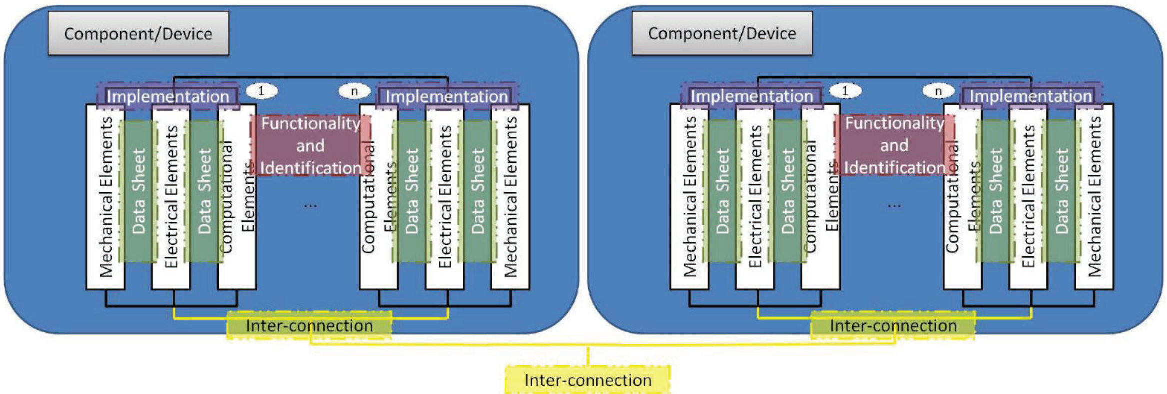


Fig. 4.5 Device perspective regarding the main aspects general service (McS)

4.2.3 Enabling technologies supporting the concepts implementation

For the implementation of the service concepts – McS and AcS – there are needed two major components in order to start programming the services. One component is represented by a SOA implementing technology in the automation field (e.g. DPWS) that is verified in a research or industrial frame. The SIRENA frame does just that, being one of the remarkable examples of SOA capabilities with the first industrial example – the dozer [7] [38].

The second major component is represented by the existence of a storing and exchange format between the control unit of a manufacturing system and the simulation. These data have a unified structure that have an integrated perspective of the production perspectives, which today, in the best case, are represented by proprietary exchange data formats. The appearance in 2006, of the neutral exchange data format – AutomationML –, that integrates under one format information, regarding not only the device topology, CAD representation, kinematics and logical states of the system (everything based on XML and IEC 62424 - CAEX), but also the possibility to attach in the future further aspects – that are also XML-based-, supports the Data Sheet aspect creation of the McS service.

Considering these two fundamentals technology pillars that are currently existing and will surely mature, we can safely formulate that it is possible and feasible the Mc Sand AcS programming in the context of the new simulation architecture proposed by the thesis. Thus, the whole manufacturing system processes and a more precise simulation can be constructed just by a succession invocation of services.

4.3 Proposed simulation architecture

The simulation architecture concept has at its basis the architecture of functionalities and the McS and AcS services that are implemented through three actors: the orchestrator, the simulation and the real manufacturing system (with its various control units).

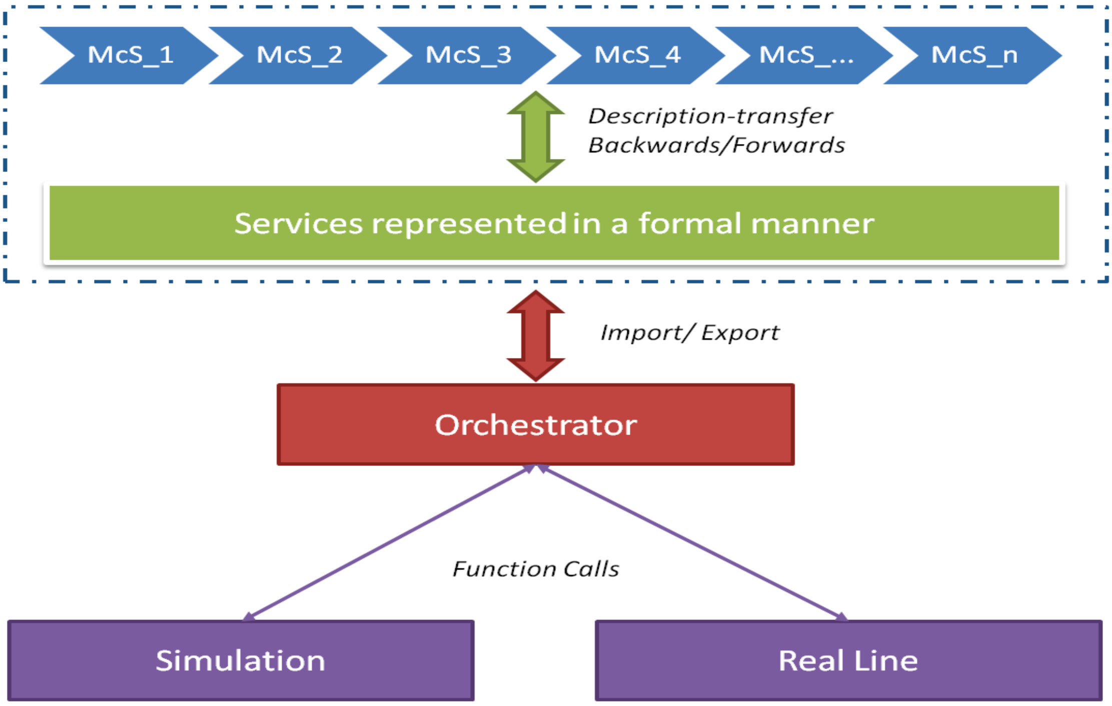


Fig. 4.10 Basic design of architecture

The first rectangle at the concept level (fig. 4.10) (represented by dot-line) represents splitting the whole manufacturing process "into" atomic sub-processes (that execute a specific functionality of the whole process) and the description transfer of their functionality –backwards or forward – under a certain formalism that should allow their verification for avoid unwanted global or sub-system states. One possible validation formalism of the sub-process state are the Petri nets. After the state validation of the whole system takes place, the validated states are embedded into services. By the service succession orchestration both simulation and manufacturing system can be controlled.

4.4 Methodology for meshing manufacturing processes

The simulation architecture proposed by the thesis is based on splitting the whole manufacturing process into "atomic" sub-processes. A splitting methodology is of great interest because till this point the division of the whole manufacturing process is done by the designers' experience.

The whole manufacturing process needs to be split into sub-processes that will be embedded into services. Generally this is done in a much customized way by the designer because until this point there are no architectural tenets how to build independent and yet interoperable and reusable services in case of manufacturing. In our methodological approach the basic procedure presented in [27] has been augmented with a verification phase. In case of a simple demonstrator deadlocks or unwanted states can be clearly identified but when dealing with complex systems it is very important to verify all aspects. The verification phase implies the service description under Petri net formalism. Petri nets are used for verification because there have a strong mathematical mechanism regarding to their creation and functioning.

The methodology (fig. 4.13) has a top-down approach. It is composed out of six sequential stages where the following should be defined:

1. *Description of the manufacturing system*: the whole manufacturing system should be formally described and its scope defined. The manufacturing system‘s subcomponents (modules) are identified and undergo identical analysis as the whole system (formal description and scope).
2. *Requirements of each module*: the identified modules (at the previous stage) are analyzed (using for example UML use-case diagram) and specific requirements are identified. At the end of this stage a set of functional units should become visible for each module.
3. *Functional units construction for each module*: at this phase all functionality needed for each module (determined by UML use-case diagram) is determined. Thus, all functional units are being defined. This may be systematically done using a Production Flow Schema (PFS) / Petri Nets (PN) approach [114] [33] [97] PFS may be used for the concept model construction for each functional unit. This could be later transferred (mapped) into a PFS activity. This in turn is refined till it reaches the necessary details to ensure proper functionality in a PN. Construction of the PN model (functional units) may utilize the transition fusion approach used by [29]. The PN constructs are very helpful because verification and functional analysis can be tested due to the mathematical basis of the PN that they are now defined by.
4. *Verification and analysis of the service (functional unit) model*: formal PN analysis techniques combined with simulation techniques (e.g. HPSim) may be used to test the models for: deadlocks, stability, existence of unforeseen prohibited states, etc.
5. *Encapsulation of functionality into components (compound functional units)*: different functional unities are encapsulated into components. This encapsulation can be done, for example, to have a closer resemblance to the hardware modules or for a more visible and understandable process.
6. *Real line validation*: Codes are written into device components / control system. Depending on the possible problems that may emerge a reiteration of the whole methodology starting from the second phase will most probably solve the issue.

Using this methodological approach the *functionality* aspect from the general service (McS) can be thoroughly determined and validated. The approach identifies also the hardware linked with the functionality aspect (Phase 1) proving valuable information to define the other general aspects (Data sheet, Interconnections, Communication, and Implementation) from within a McS. Furthermore, applying this methodological approach to various manufacturing systems typologies would lead to general guiding lines and/or pre-defined/basic services manufacturing systems typologies would lead to general guiding lines and/or pre-defined/basic services

Fig. 4.13 Methodological approach for the functional units delimitation and verification

5 Conception of the prototype architecture

In this chapter there will be a detailed description of the implementation of the basic simulation architecture using current material flow simulation (Plant Simulation 9) and controllers (PLC).

5.2 Architectural concept implemented into the prototype

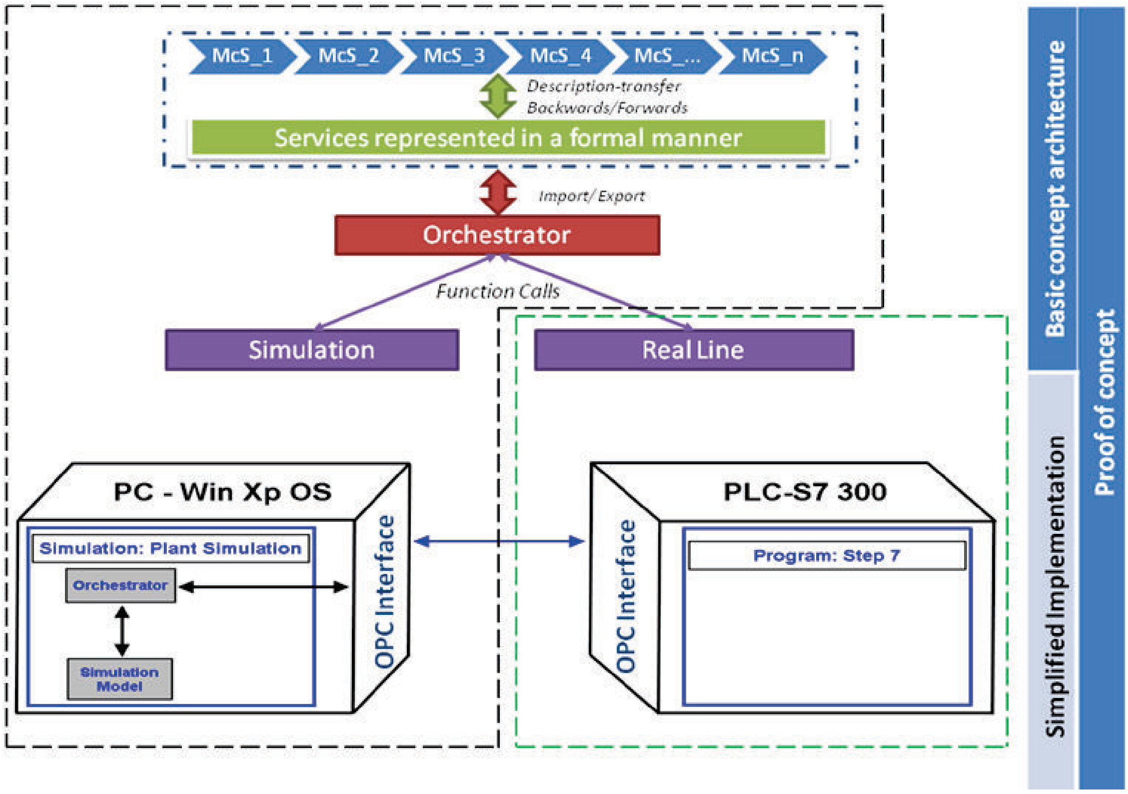


Fig. 5.3 Architectural concept implemented into the prototype

As shortly presented into chapter 4.3, the proposed simulation architecture (fig. 5.3) has at its basis the architecture of functionalities and the McS and AcS services that are implemented through three actors: the orchestrator, the simulation and the real manufacturing system. The first rectangle at the concept level (fig. 5.3) (represented by dot-line) represents splitting (meshing) of the whole manufacturing process “into” atomic sub-processes (that execute a specific functionality of the whole process) and the description transfer of their functionality under a certain formalism. After the state validation of the whole system takes place, the validated states are embedded into services. By the service succession orchestration both simulation and manufacturing system can be controlled.

The prototype implementation reduces the simulation architecture to only two entities: the simulation and the manufacturing system. As depicted from picture nr. 5.3, at this prototype simulation architecture, in the simulation environment the manufacturing splitting process takes places and also the orchestration and simulation model are residing here. Obviously the implementation of the basic architecture (in the upper side of fig. 5.3) could have a more complex implementation like in figure 7.1 or 7.2. Thus, the basic architecture remains the same, and its implementation may vary depending on the available technologies. Current technological limitations imposed the approach presented into figure5.3.

To summarize, the first prototype implementation of the architecture has the following characteristics:

- Orchestration and model simulation reside both in the simulation environment (Plant Simulation 9). The orchestration will coordinate the simulation model and the control unit (PLC) of the real manufacturing system;
- Testing of the sub-process states will not takes place due to the reduced number of sub-processes that compose the manufacturing systems’ functionality;
- Services which in the general concept were shaped into McS and AcS, are replaced by objects within the simulation and the service calls are replaced by function calls (allowed by the OPC specification) between the simulation and the control unit of the manufacturing system. McS and AcS programming is not the target of the thesis, because their development has a high degree of implementation issues which was estimated beyond a two year period taking into consideration a twenty people programming team. Another limitation is the lack of OPC UA implementation at the software applications level, both in simulation and control unit, in order to exchange more real process relevant data between the two;
- The control unit of the manufacturing line will use a S7-300 PLC controller manufactured by Siemens, largely adopted by the manufacturing industry;
- The simulation environment – Plant Simulation 9 – will operate on Microsoft Windows XP operating system to benefit from the implemented OPC application interaction capabilities based on the (D)COM technology.

5.3 Basic simulation architecture concept implementation

5.3.1 Description of the Mobile Module manufacturing system

Fig. 5.4 Mobile Module [58]

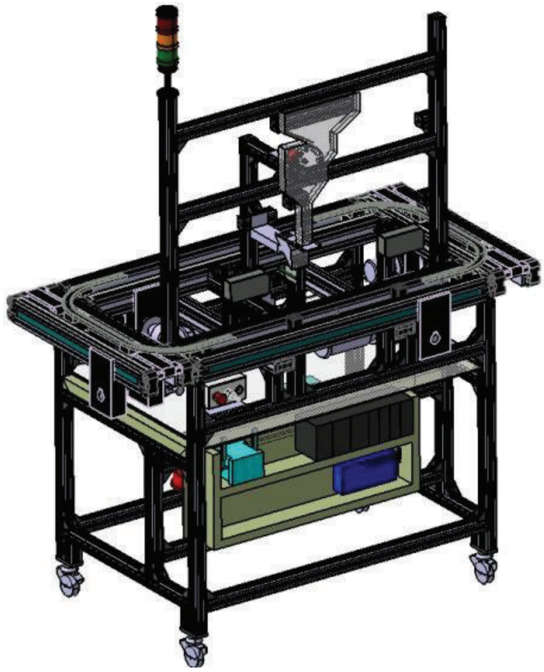


Fig. 5.5 CAD model of Mobile Module [80]

The manufacturing system that was used to implement the simulation architecture is named the "Mobile Module", which represents one unit from SmartFactory^{KL} platform (Kaiserslautern), where the new and future technologies are researched.

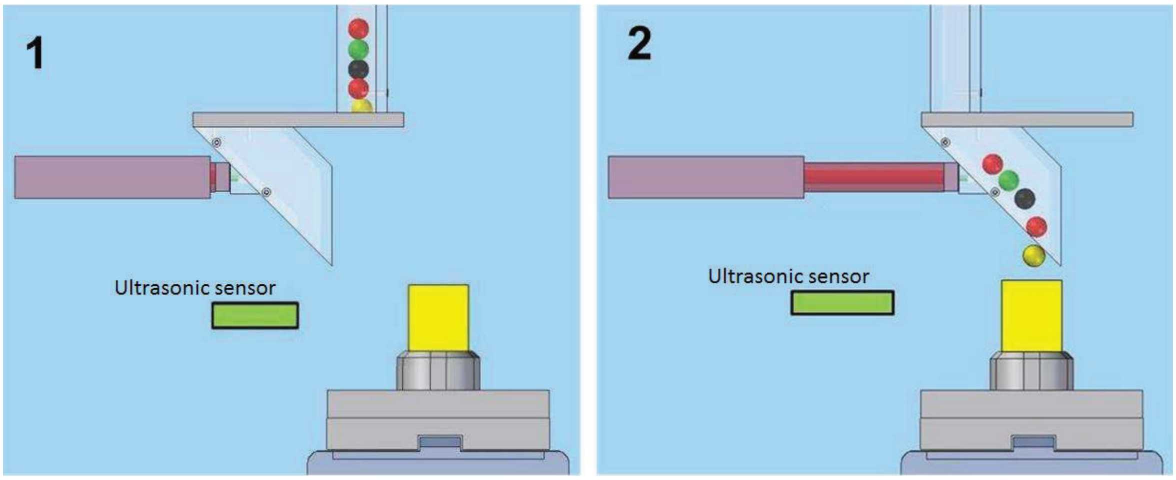


Fig. 5.8 Loading process [59]

The functionality of the manufacturing system is to load pills into boxes (located on transporters) that move on a conveyer (fig. 5.5) – built by the GRASSLIN company (approximately one meter high, having rectangular shape composed out of four straight transporting belts). The number of pills is read from an RFID tag which is glued on the pill box, with the read-transponder. After a pill box is loaded, on the RFID tag is written that the box is full with the second transponder on the line: the write transponder. In the loading process the pills are manipulated with a wheel that has special places where only one pill fits. The pills which are dropped from the wheel in the pre-loading pill buffer are counted with a light barrier attached to the source's structure (fig 5.8, 5.10).

The pre-loading pill buffer will open only if the pill box is still present on the container in the loading position. The presence of the box that has to be filled is sensed by an ultrasonic sensor. The containers are sensed with inductive sensors that activate or deactivate the stoppers. There can be only one container in the loading area; all others will wait until that container will leave the loading area. Locations of all sensors and actuators can be identified in figure 5.11 (upper view).

Description of the complete process executed by the Mobile Module is thoroughly represented by figure 5.12

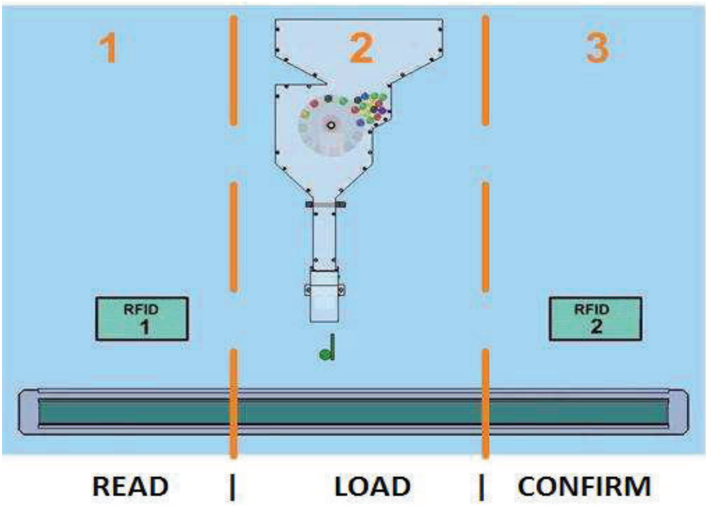


Fig. 5.9 Mobile Module functionality [59]

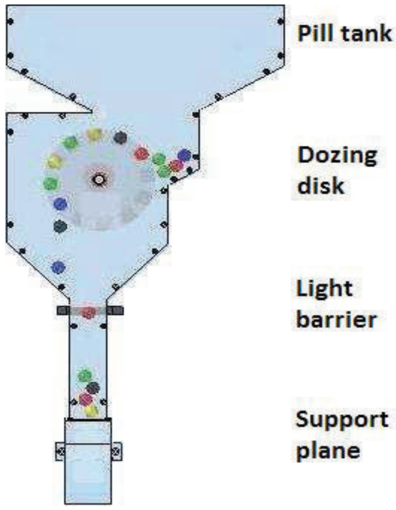


Fig. 5.10 Dozer construction [59]

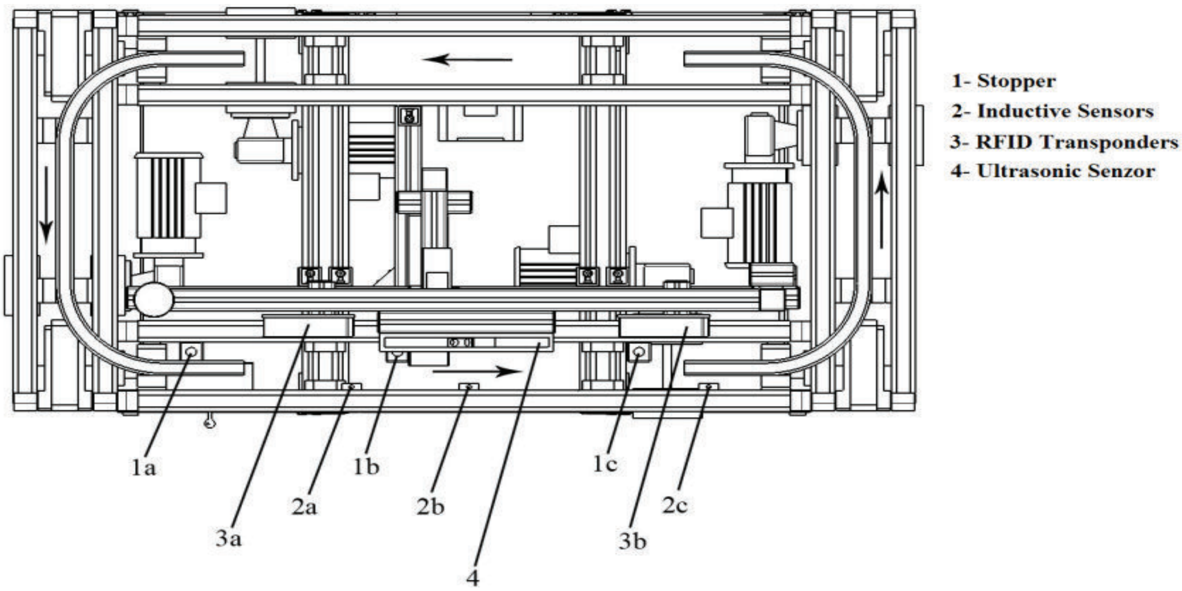


Fig. 5.11 Actuators and sensors of the Mobile Module (upper view) [80]

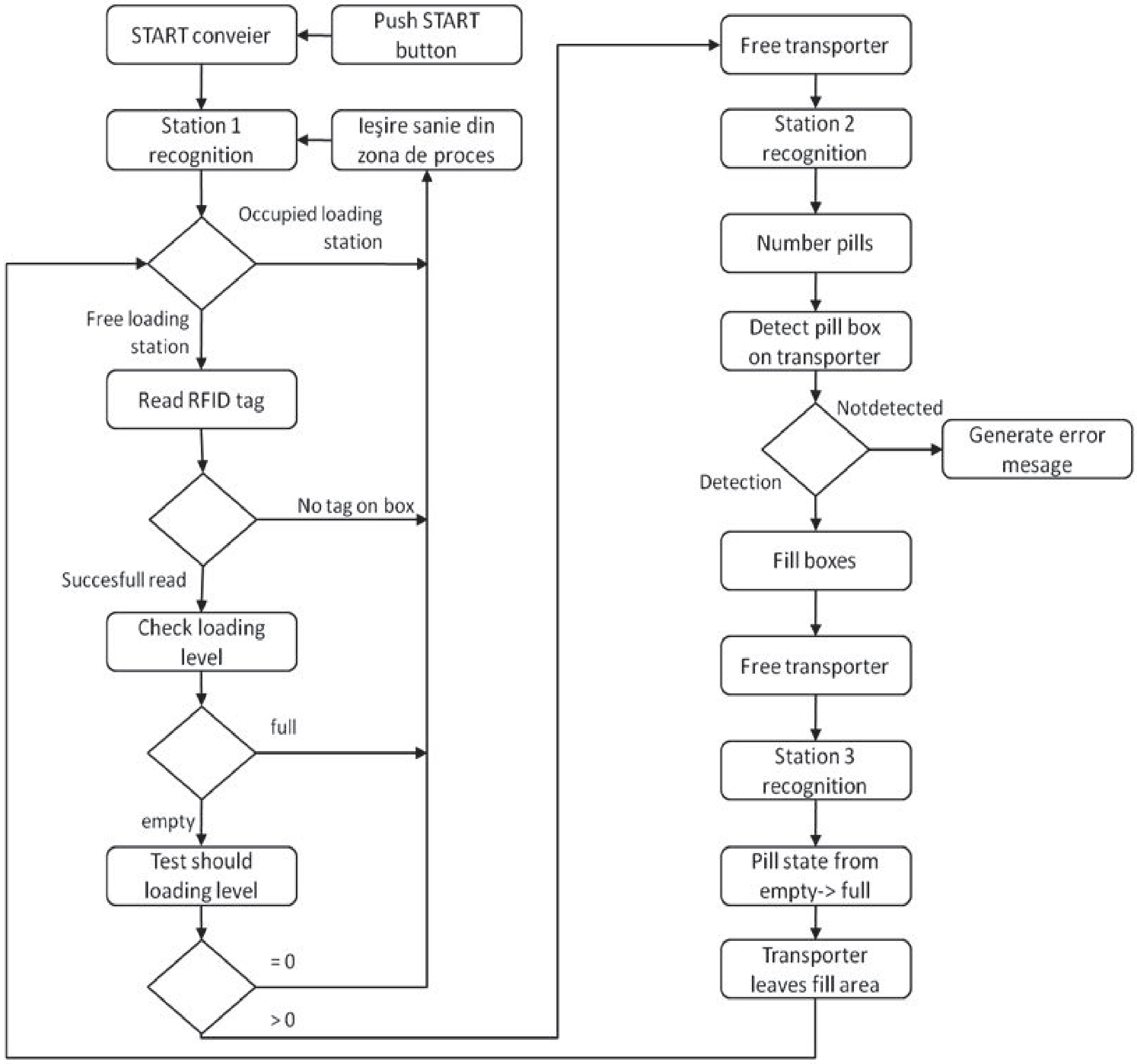


Fig. 5.12 Mobile Module process description

5.3.2 Software package used for orchestration and simulation modeling

Discrete event simulation (DES) is a branch of the simulation methodology largely used in industry to analyze manufacturing systems, and such a methodology is used by Plant Simulation 9. The DES software Plant Simulation is part of the Siemens PLM package and allows the creation of digital simulation models of logistic systems (e.g. manufacturing systems) that helps explore the system characteristics and optimize its parameters. These digital models allow running experiments and what-if scenarios without disturbing existing production systems or – when used in the planning process – long before the real production systems are installed. The results provide the information needed to make fast, reliable, smarter decisions in the early stages of production planning [144].

Using Plant Simulation manufacturing systems can be modelled and their processes simulated. In addition, it can optimize material flow, resource utilization and logistics for all levels of plant planning from global production facilities, through local plants, to specific lines [144].

5.3.3 Main implementation aspects of the simulation architecture

The Plant Simulation digital model includes both the orchestration and the simulation model of the Mobile Module (with lower detail level are represented the Siemens, SAP modules, present at the 2009 CeBIT event, by the Siemens and SAP frames). The separation of the orchestration from the simulation model is done by creating two distinct frames: one corresponding to the global aspects orchestration – that enables the global functionality -, and, the second frame, that comprises the "local/internal" behavior of each important object of the simulation model – that enables the internal functionality.

The basic architectural implementation (fig. 5.13), with the simulation model done in Plant Simulation, follows the SOA concept tenets – independence and interoperability between entities. Thus, the simulation model construction strives at creating independent and interoperable entities, and the process succession to be achieved by orchestration. The date exchange between simulation and control unit is implemented at the orchestration level by an OPC interface.

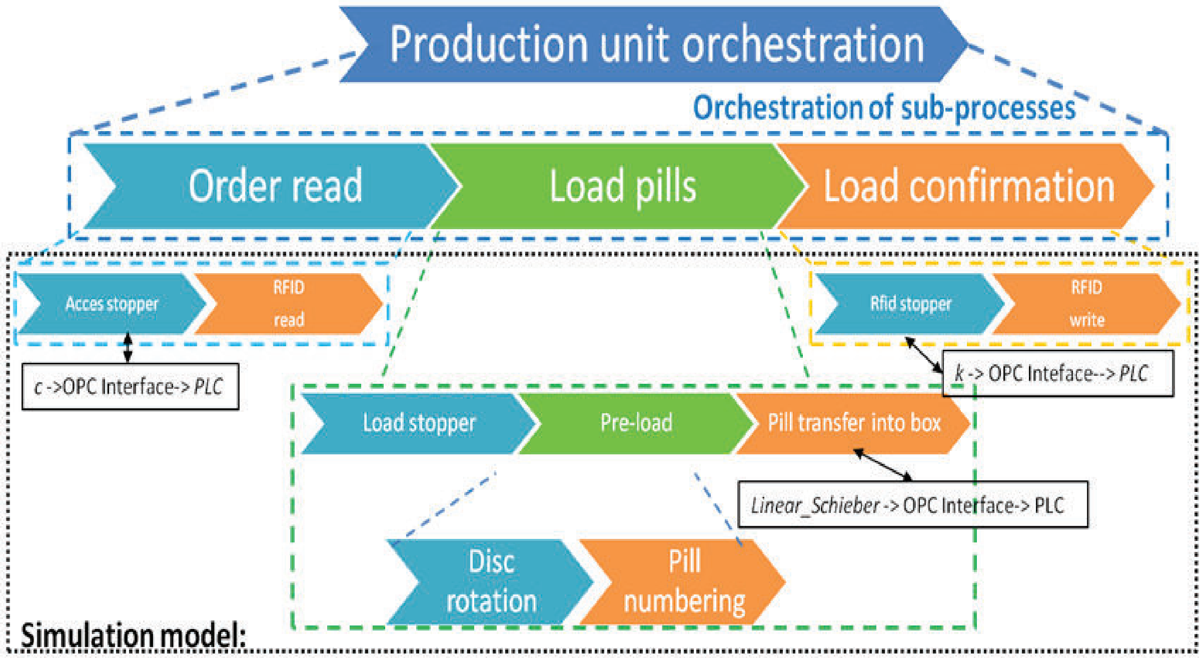


Fig. 5.13 Simulation architecture implementation into the first prototype

The simulation model orchestration is done with the logic implemented into the *Logik* frame by the contained objects, and obviously by the discrete event engine of Plant Simulation. The *Logik* frame represents the bounding element between the simulation model – represented by the objects from the *SF_mobile_module* – that executes the global level logic. The internal/local logic is done mainly by the objects existing in the following frames: *RFIDtagREader*, *Pillsource*, *pillalimentation*, *RFIDtagWriter* toghether with the *Line* object and the movable units (*aluminiumcontainer*, *pill_container*, *pillbox*, *pill*).

The orchestration focuses mainly the interaction logic at the filling process (fig. 5.9). The methods that represent the interaction logic between the objects located in the simulation model (*SF_mobile_module*) are triggered mainly by sensors located at the *Line* object level (e.g. *sensor2*, *sensor4*, *sensor7* methods etc.). Besides these methods, in case of the orchestration, we have also the global variables. They play a crucial role in the interaction between the objects at the simulation model, but, extremely important, are the global variables (e.g. *Linear_Schieber*, *k*, *c* etc.) that trigger the PLC state change correlated with the events taking place into the simulation.

6 Final conclusions. Main thesis contributions

6.1 Final conclusions

The final research conclusions can be summarized as follows:

- The Digital Factory concept can be applied at the industrial level on a large scale; the benefits of the entire concept implementation lead to more efficient manufacturing systems and a smoother migration of optimized manufacturing scenarios, from simulation to manufacturing systems (and vice-versa), can be achieved;
- Currently the Digital Factory concept is in an early implementation phase, characterized by the lack of all simulation tools integration (e.g. Siemens PLM) and also a reduced level of application interaction between manufacturing simulation and manufacturing systems, that is currently executed by the OPC specification;
- Improvement of Digital Factory concept implementation could be done by integrating all simulation and manufacturing applications by the OPC UA specification – the IEC 62541 standard; also, the implementation time of an optimum manufacturing scenario at the manufacturing level can be significantly reduced by using distributed control units (e.g. PLC in compliance with IEC 61499);
- The OPC UA specification proves versatility not only at the transport level, with its two transport protocols (one based on Web services and the other based on an optimized binary TCP protocol), but also at the informational level modeling with meta-modeling capabilities. Although it reaches lower transfer speeds, compared to OPC, this disadvantage can be overcome be the coexistence characteristics of OPC UA with OPC;
- OPC UA is not yet fully implemented (in the best case partially) at the manufacturing simulation level. The classic OPC specification is still the middleware solution adopted by the major engineering simulation packages (Delmia, Siemes PLM etc.);
- Service-oriented architectures applied to the automation level, that implies the development of independent and interoperable modules (software and hardware), in order to allow a flexible and simpler approach in case of manufacturing reconfigurability, using currently as implementation technologies the Web-based approaches (e.g. WS, DPWS, UPnP), are gaining more interest and attention in the research and industrial field;
- Encapsulation at the service level, not just of functional and control aspects regarding devices but also the correlation and augmentation of the fore mentioned ones with relevant simulation aspects, represents the thesis proposal for the Manufacturing-centred Service – McS;
- In the perspective of achieving the ultimate SIRENA goals, that is to apply SOA to the lowest levels of the automation pyramid (field devices), the Automation-centred service – ScA – represents a “light” aspect structure of the McS, in order to be hosted

by future computational capable field devices and to stand alone assure the manufacturing systems functionality;

- A new simulation architecture is proposed, that is based on: the architecture of “functionalities” – that presumes splitting (meshing) of the entire manufacturing process into “atomic” sub-processes depending on the process and technological restrictions, the usage of the McS and AcS services for the atomic sub-processes description at the simulation and controller level, and the control mechanism, of both simulation and manufacturing system, by the orchestration of the service succession. Thus, in the thesis vision, an implementation improvement of the Digital Factory, respectively its integration with manufacturing systems, can be achieved.
- A first prototype is developed, to integrate the simulation architecture, using current technologies from the manufacturing simulation (Plant Simulation 9), middleware (OPC) and control units (S7-300 Siemens controller). The prototype demonstrates also the current simulation approaches limitations.
- Manufacturing systems modeling using just mathematical methods is not a viable solution because of the modeling complexity, reduced scalability, intractability and modeling time; as recommendations mathematical methods should be applied to “narrow” areas where process bottlenecks are earlier determined by other methods;
- As optimal accepted manufacturing modeling solution by industry is the discrete-event simulation technique besides an object-oriented derived method.

6.2 Original contributions

By this PhD thesis original contributions have been brought regarding the Digital Factory modeling and implementation into real manufacturing systems, contributions published or to by published by the author ([54], [77], [78], [79], [80], [81]), inland or abroad, as follows:

Regarding the theoretical standpoint:

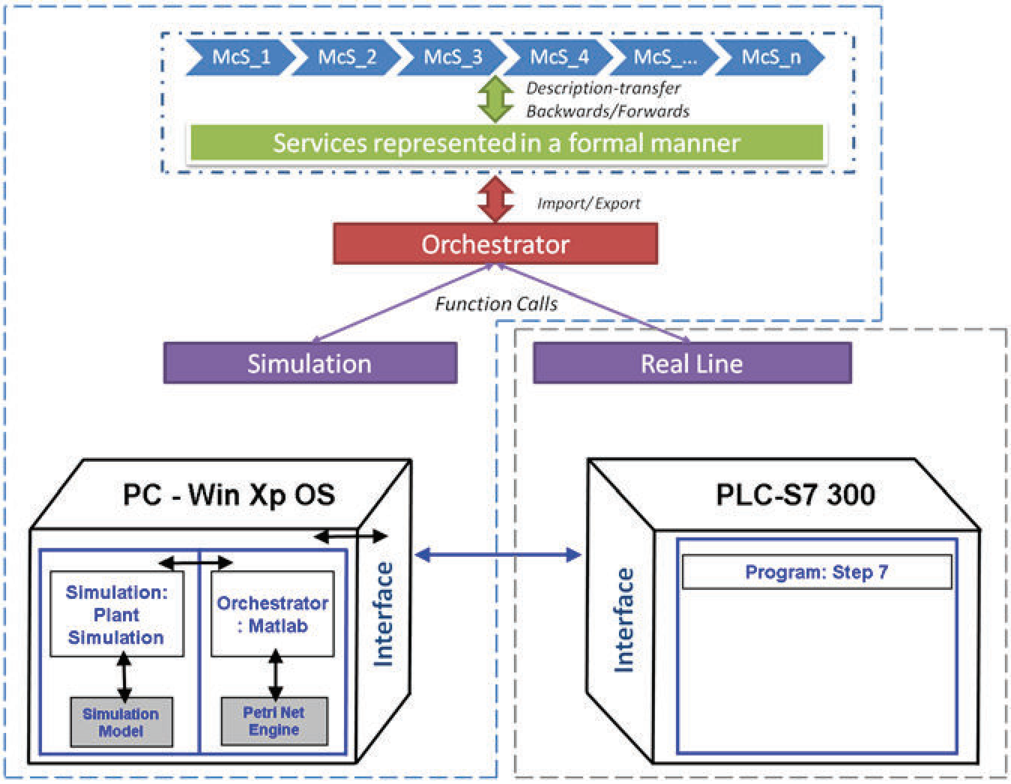
- Most scientific and technical published results from this field have been synthesized as a bibliographical study;
- An analysis has been performed concerning where manufacturing simulation can be used independently, together and separately from the manufacturing system. Two major cases resulted that recommended the development of the McS and AcS - services;
- The augmentation necessity was formulated regarding the current SOA approaches (applied in automation) with specific simulation elements. The augmentated service is theoretically defined and is called Manufacturing-centered Service – McS.
- Supporting the development of field devices with computational capabilities, the Automation-centred service was theoretically defined, service which encapsulates all necessary aspects to allow the complete functionality of an automated manufacturing system;

- The technological premises, AutomationML and the SIRENA frame, where presented to support the McS and AcS implementation;
- The architecture of "functionalities" concept was formulated in order to "mesh" the whole manufacturing system – depending on the technical and process limitation, into "atomic" sub-processes;
- A possible meshing methodology was presented to split the entire manufacturing processes into "atomic" sub-processes and to verify the system states after the re-composition of the entire manufacturing process by means of "atomic" processes;
- A new simulation and manufacturing simulation architecture was proposed that incorporates the architecture of "functionality" and McS and AcS service concepts, that allows primarily the construction and coordination of both simulation and manufacturing system, but also could verify the system states after the global "atomic" units functional composition.

Regarding the practical standpoint:

- A first simulation architecture prototype was built. Thus, at the simulation level (Plant Simulation 9) the orchestration and simulation model were integrated, and the real manufacturing system was controlled from the simulation by the OPC interface;
- A better control possibility from the simulation was demonstrated, concerning more actuators belonging to the manufacturing system (Mobile Module), by using the standard OPC interface;
- A CAD model in Catia v5 was constructed, so that the simulation animation in 3D was closer to reality;
- A synthetic documentation was conceived regarding the steps that need to be followed to configure and verify the OPC server and client;
- A usability test was conducted regarding the OPC interface from Plant Simulation 9, and improvement solutions were proposed to Siemens PLM;

7 Research outlook



7.1 First improvement scenario

The main research direction is focused towards an improved implementation of the simulation architecture, but also the development of manufacturing processes "splitting" approach.

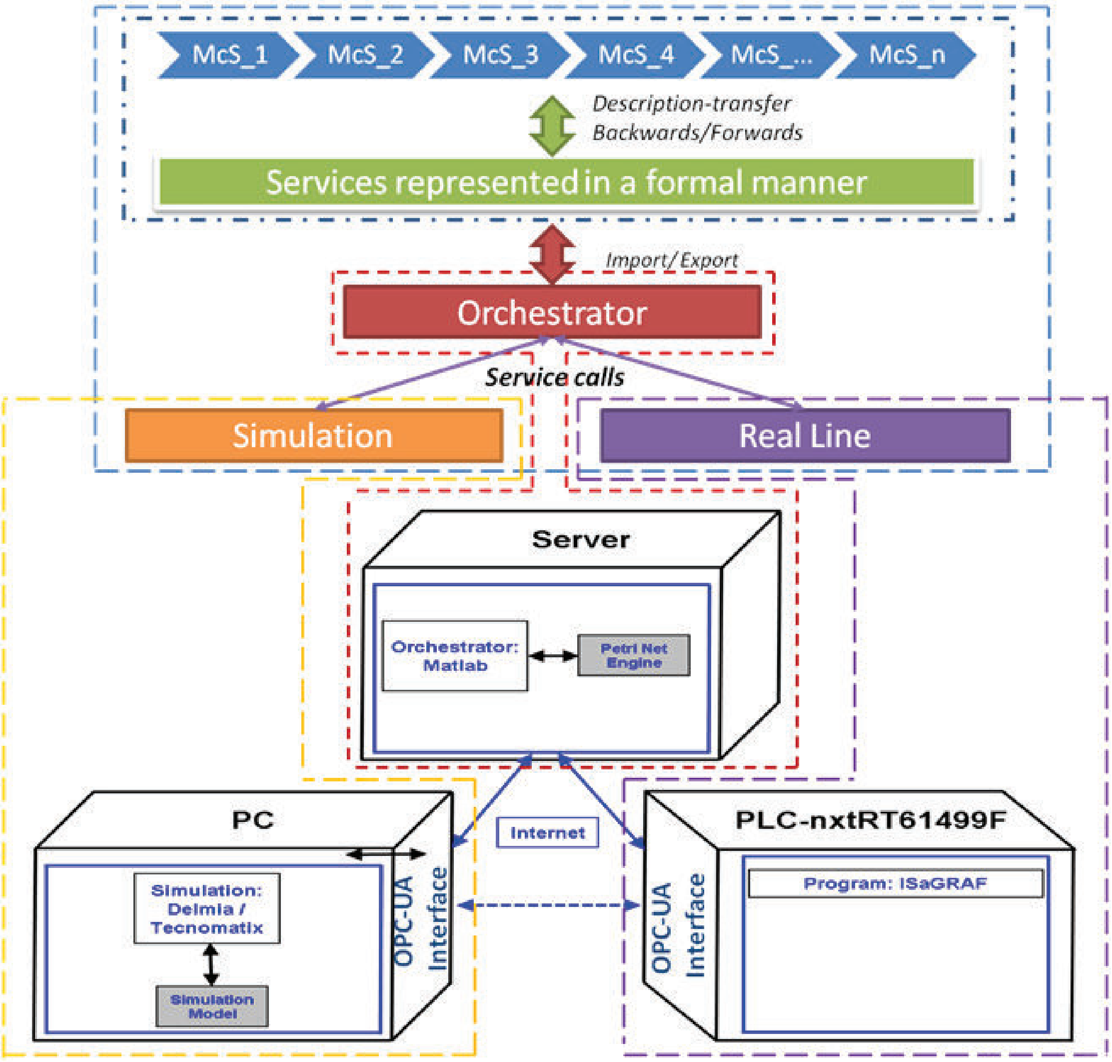
A first simulation architecture implementation improvement would be the orchestration outside (fig. 7.1) of the simulation environment (same host or a different host accessible via LAN connection). This could be achieved by the "atomic" process description by means of Petri nets. Thus, possible system states could be verified, and the orchestration may be done by a Petri net engine (possibly in Matlab) outside of the manufacturing simulation. Linkage between simulation and PLC will be done by an OPC interface. Still, at this phase a process data exchange between simulation and controllers is not yet possible due to the lack of a better middleware specification implementation.

A second major improvement step would be the encapsulation of the McS and AcS aspects as software implementation. The current vision is to use the DPWS technology, the currently most promising Web-based technology, and, to integrate this, with the neutral exchange format AutomatioML. Thus, both the migration and execution of both simulation and manufacturing system could be further improved. Application interaction, between orchestrator, simulation and control unit, will most likely imply OPC UA, because of the extension of the needed information exchange, compared to current implemented middleware approaches (mainly OPC).

One of the last phases would be to integrate the last two major steps: orchestration outside of the simulation and a real process data exchange between the simulation and the control unit using as communication infrastructure the Internet (fig. 7.2). Thus, the hardware implemented into the simulation architecture could have an evolved representation like the IEC 61499 controllers or a microcontroller-based approach (fig.7.2). As simulation environment both Delmia and Tecnomatic platforms could be used, and, as communication infrastructure may be used the Internet.

The final objective would be to apply the architecture till the field device level, thus, a one to one correlation between the physical representation and functionality. So, the level of abstraction would be identical both in simulation and regarding the simulation.

Research regarding a manufacturing processes "splitting" approach will be also researched, in order to determine the means to recognize, reuse and standardize the "atomic" sub-processes and respectively the corresponding services.



7.2 Future implementation scenario

Bibliography

1. Ajmone Marsan, M., Balbo, G., Bobio, A., Chiola, G., Conte, G., & Cumani, A. (1985). *On Petri nets with stochastic timing.* Int. Workshop on Timed Petri Nets, Torino.
2. Aldinger, L., Rönnecke, T., Hummel, V., & Westkämper, E. (2006). Advanced Industrial Engineering. *Industrie Management , 22*, 59-62.
3. Ari, R.-A. (2009). *Research report: Plant Simulation modeling of the Continuous Process Part of the SmartFactoryKL Demonstration plant.* University of Kaiserslautern, Kaiserslautern.
4. Banks, J., Carson, J., & Nelson, B. (1996). *Discrete-Event Simulation.* Upper Saddle River, NJ: Prentice Hall.
5. Blanco, P., Poli, M., & Barretto, M. (2003). OPC and CORBA in manufacturing execution systems: a review. *Emerging Technologies and Factory Automation (ETFA). 2*, pp. 50-57. Lisbon: IEEE.
6. Bley, H., & Franke, C. (2001). Integration von Produkt- und Produktionsmodell mit Hilfe der Digitalen Fabrik. *Werkstattstechnik online , 91*, 214-220.
7. Bohn, H., Bobek, A., & Golatowski, F. (2006). SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains. *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies* (pp. 43-48). Tahiti – Moorea: IEEE.
8. Bondrea, I. (1998). *Modelarea şi simularea proceselor de producţie.* Sibiu: Edituta Universităţii din Sibiu.
9. Bratley, P., Fox, B., & Schrage, L. (1987). *A Guide to Simulation.* New York: Springer.
10. Budszuhn, F. (2008). *Visual C++ Windows-Programmierung mit den MFC* (Vol. 5). München, Germany: Addison Wesley.
11. Bullinger, H.-J., Spath, D., Warnecke, H.-J., & Westkämper, E. (2009). *Handbuch Unternehmensorganisation - Strategien, Planung, Umsetzung.* Berlin Heidelberg: Springer.
12. Busenberg, S., Forte, B., & Kuiken, H. K. (1991). Mathematical Modelling of Industrial Processes. Berlin: Springer.
13. Cachapa, D., Colombo, A., Feike, M., & Bepperling, A. (2007). An Approach for Integrating Real and Virtual Production Automation Devices Applying the Service-oriented Architecture Paradigm. *12th IEEE Conference on Emerging Technologies and Factory Automation* (pp. 309-314). Patras: IEEE.
14. Cao, X., & Ho, Y. (1990). Models of discrete event dynamic systems. *IEEE Control Systems Magazine , 10*, 69–75.
15. Cassandras, C., Lafortune, S., & Olsder, G. (1995). *Introduction to the modelling,control and optimization of discrete event systems.* (A. Isidori, Ed.) Springer.
16. Cohen, G., Gaubert, S., & Quadrat, J. (. (2001). *Max-plus algebra in Scilab and applications.* Workshop, Pekin.
17. Dashchenko, A. I. (2006). *Reconfigurable manufacturing systems and transformable factories.* Berlin Heidelberg: Springer.
18. Desrochers, A., & Al-Jaar, R. (1993). *Petri Nets for Automated Manufacturing Systems: Modeling, Control and Performance Analysis.* Troy, New York: Rensselaer Polytechnic Institute.
19. Dombrowski, U., Tiedemann, H., & Bothe, T. (2001). Visionen für die Digitale Fabrik. *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZwF) , 96*, pp. 96-115.
20. Drath, R. (2010). *Datenaustausch in der Anlagenplanung mit AutomationML* (1 ed.). Berlin, Heidelberg, Germany: Springer.
21. Drath, R., Luder, A., Peschke, J., & Hundt, L. (2008). AutomationML - the glue for seamless automation engineering. *13th International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 616-623). Hamburg: IEEE.

22. Erl, T. (2010). *Service-Oriented Architecture (SOA): Concepts, Technology, and Design* (10th ed.). Upper Saddle River, New Jersey, United States of America: Prentice Hall.
23. Fahrwinkel, U. (1995). *Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering.* Paderborn : Universität Paderborn .
24. Ferstl, O. K., & Sinz, E. (1993). *Grundlagen der Wirtschaftsinformatik.* Oldenbourg, München, Wien.
25. Frank, E. (2009, 01). Verteilte Steuerungen nach IEC 61499.
26. Freedman, P. (1991). Time, Petri nets, and robotics. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION , 7* (4), 417-433.
27. Garcia Melo, J., Souit, S., Santos Filho, D., Junqueira, F., & Miyagi, P. (2010). A systematical approach to expose manufacturing system as a service. *9th IEEE/IAS International Conference on Industry Applications* (pp. 1-6). Sao Paulo: IEEE.
28. Garro, O. M. (1993). Towards New Architecture of Machine Tools. *International Journal of Production Research , 31*, 2403-2414.
29. Gausemeier, J., & Stollt, G. (2006). Eine Systematik zur Gestaltung der Produktion von morgen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZwF)* , 28-34.
30. Gomes, L., & Barros, J. (2005). Structuring and composability issues in Petri nets modeling. *IEEE Transactions on Industrial Informatics , 1* (2), 112-123.
31. Hagge, N., & Wagner, B. (2005, Mai). A New Function Block Modeling Language Based on Petri Nets for Automatic Code Generation. *IEEE Trans. on Industrial Informatics , 1* (4), pp. 226-237.
32. Hanisch, H.-M., Hirsch, M., Missal, D., Preuße, S., & Gerber, C. (2009). One Decade of IEC 61499 Modelling and Verification - Results and Open Issues. In N. Bakhtadze, & A. Dolgui (Ed.), *13th IFAC Symposium on INFORMATION CONTROL PROBLEMS IN MANUFACTURING*, (pp. 211-216). Moscow.
33. Hasegawa, K., Takahashi, K., & Miyagi, P. (1988). Application of the Mark Flow Graph to represent discrete event production systems and system control. *Transactions of the Society of Instrument and Control Engineers , 24* (1), 67-75.
34. Ibrahim, N. (2009). Orthogonal Classification of Middleware Technologies. *Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (pp. 46-51). Sliema: IEEE.
35. Ionescu, R. (2007, Decembrie). *Universitatea Transilvania din Brasov*. Preluat pe Februarie 10, 2010, de pe http://cciu.unitbv.ro/files/RETELE DE COMUNICATII DATE - subset.pdf
36. Jammes, F. (2011). Real time device level Service-Oriented Architectures. *IEEE International Symposium on Industrial Electronics (ISIE)* (pp. 1722-1726). Gdansk: IEEE.
37. Jammes, F., & Smit, H. (2005). Service-oriented architectures for devices - the SIRENA view. *3rd IEEE International Conference on Industrial Informatics (INDIN)* (pp. 140-147). Perth: IEEE.
38. Jammes, F., & Smit, H. (2005, February). Service-Oriented Paradigms in Industrial Automation. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS , 1* (1), pp. 62-70.
39. Jammes, F., Smit, H., Lastra, J., & Delamer, I. (2005). Orchestration of service-oriented manufacturing processes. *10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA). 1*, pp. 617-624. Catania: IEEE.
40. Janson, F., & Zetterquist, M. (2000). *CORBA vs. DCOM.* Master's Thesis in Computer Science, The Royal Institute of Technology Kungliga Tekniska Högskolan, Stockholm.
41. Kelton, W. S., & Sadowski, D. (2002). *Simulation with Arena – 2nd ed.,.* McGraw Hill Companies.
42. Kirkham, T., Savio, D., Smit, H., Harrison, R., Monfared, R., & Phaithoonbuathong, P. (2008). SOA MIDDLEWARE AND AUTOMATION: Services, Applications and Architectures. *6th IEEE International Conference on Industrial Informatics* (pp. 86-91). Daejeon: IEEE.
43. Koren, Y. (1983). *Computer Control of Manufacturing Systems.* New York: McGraw Hill.
44. Koren, Y. (2010). *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems.* New Jersey: John Wiley&Sons.

45. Koren, Y., Jovane, F., Heisel, U., Moriwaki, T., Pritschow, G., Ulsoy, A., et al. (1999). Reconfigurable Manufacturing Systems (Also a Keynote paper presented at the General Assembly). *Annals of the CIRP*, *48*, pp. 6-12.
46. Krause, F.-L. (2001). Digitale Fabrik. *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZwF) , 96*, 84.
47. Kühn, W. (2006). *Digitale Fabrik - Fabriksimulation für Produktionsplaner.* München Viena: Cart Hanser Verlag.
48. Langemann, T. (1999). *Modellierung als Kernfunktion einer systemorientierten Analyse und Bewertung der diskreten Produktion.* Paderborn: HNI-Verlagsschriftenreihe.
49. Leitao, P., Mendes, J., & Colombo, A. (2009). Smooth migration from the Virtual design to the real manufacturing control. *7th International Conference on Industrial Informatics* (pp. 539-544). Cardiff: IEEE.
50. Leurs, L. (2009). *Hilfsblätter zur Vorlesung - Industrielle Steuerungstechnik .* Kaiserslautern: Bosch Rexroth.
51. Levi, S., & Agrawala, A. (1990). *Real Time System Design.* McGraw–Hill.
52. Lewis, F., Huang, H., Pastravanu, O., & Gürel, A. (1995). Control systems design for flexible manufacturing systems. În A. Raouf, & M. Ben-Daya, *Flexible Manufacturing Systems: Recent Developments* (pg. 253-290). Elsevier Science.
53. Lewis, R. (2001). *Modelling control systems using IEC 61499: applying function blocks to distributed systems.* London, UK: The Institution of Electrical Engineers.
54. Loskyll, M., Schlick, J., Hodek, S., Ollinger, L., Gerber, T., Pîrvu, B. (2011). Semantic Service Discovery and Orchestration for Manufacturing Processes. *16th IEEE International Conference on Emerging Technologies and Factory Automation* (pp. 1-8). Toulouse: IEEE.
55. Lüder, A., Hundt, L., & Keibel, A. (2010). Description of manufacturing processes using AutomationML. *IEEE 15th Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1-8). Bilbao: IEEE.
56. Mahnke, W., Leitner, S., & Damm, M. (2009). *OPC Unified Architecture.* Berlin Heidelberg: Springer - Verlag.
57. Marco, M., Leitao, P., Colombo, A., & Restivo, F. (2008). Service-Oriented Control Architecture for Reconfigurable Production Systems. *IEEE International Conference on Industrial Informatics* (pp. 744-749). Daejeon: IEEE.
58. Marin, R. (2009). *Research report: Simularea în "Plant Simulation" a modulului mobil smartFactoryKL şi realizarea unei conexiuni printr-un server OPC.* University of Kaiserslautern, Kaiserslautern.
59. Meierer, N. (2011). *Konzeption und Programmierung einer PC basierten Steuerung von industrieüblichen Feldgeräten am Beispiel eines Stückgut-Abfüllprozess.* Studienarbeit, Technische Universität Kaiserslautern, Kaiserslautern.
60. Meyer, B. (1998). *Objektorientierte Software-Entwicklung.* München: Hanser.
61. Mortensen, K. (2003). *http://www.informatik.uni-hamburg.de/TGI/PetriNets/*. Retrieved August 30, 2010, from Uni-Hamburg: http://www.informatik.uni-hamburg.de/TGI/PetriNets/
62. Müller, S., Carnevale, M., & Zäh, M. F. (2003). Roadmap zur Virtuellen Produktion. In M. F. Zäh, & G. Reinhart, *Virtuelle Produktion – Nutzenpotenziale im Lebenszyklus der Fabrik* (pp. 1-1 -1-21). München: Utz.
63. Nissanke, N. (1997). *Real Time Systems.* New Jersey: Prentice Hall.
64. Norman, R. J. (1998, May). *CORBA and DCOM: Side by Side.* Retrieved March 25, 2011, from San Diego State University: http://www-rohan.sdsu.edu/faculty/rnorman/papers/corba-dcom.pdf
65. Ohmer, M. (2011). *Einsatz von UPnP und DPWS in einer serviceorientierten Produktionsanlage.* Studienarbeit, University of Kaiserslautern, Kaiserslautern.

66. Ollinger, L., Schlick, J., & Hodek, S. (2011). Leveraging the Agility of Manufacturing Chains by Combining Process-Oriented Production Planning and Service-Oriented Manufacturing Automation. *Proceedings of the 18th IFAC World Congress (to appear).* Milano.
67. Paolucci, M., & Sacile, R. (2005). *Agent-based manufacturing and control systems : new agile manufacturing solutions for achieving peak performance.* Boca Raton, Florida: CRC PRESS.
68. Park, K.-J., & Lee, Y.-H. (2004). An On-line Simulation Approach to Search Efficient Values of Decision Variables in Stochastic Systems. *The International Journal of Advanced Manufacturing Technology , 25*, 1232-1240.
69. Park, Y., Kirkham, T., Phaithoonbuathong, P., & Harrison, R. (2009). Implementing agile and collaborative automation using Web Service orchestration. IEEE International Symposium on Industrial Electronics (pp. 86-91). Seoul: IEEE.
70. Parunak, H. D., Savit, R., & Riolo, R. L. (2008). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. Proceedings of Multi-agent systems and Agent-based Simulation (pp. 1-16). Paris: Springer.
71. Păstrăvanu, O., Matcovschi, M., & Mahulea, C. (2002). *Aplicaţii ale reţelelor Petri în studierea sistemelor cu evenimente discrete.* Iaşi: Editura „GH. ASACHI”.
72. Peterson, J. (1981). *Petri Net Theory and the Modeling of Systems.* Prentice–Hall.
73. Petri, C. (1962). *Kommunikation mit Automaten.* Technical Report, Institut für Instrumentelle Mathematik, Bonn.
74. Pfeifer, T., & Schmitt, R. (2006). *Autonome Produktionzellen - Komplexe Produktionsprozesse flexibel automatisieren.* Berlin Heidelberg: Springer.
75. Pîrvu, B. (2009). *Integration weiterer SPS Code Variablen bspw. Lampe, Stopper und Näherungsinitiatoren.* Studienarbeit, Technische Universität Kaiserslautern, Kaiserslautern.
76. Pîrvu, B. (2011). *Modelarea sistemelor de producţie.* Raport Ştiinţific, Facultatea de Inginerie "Hermann Oberth", Sibiu.
77. Pîrvu, B., & Bondrea, I. (2011). Motivation behind a new Approach that links that links Manufacturing Simulation with future controllers. *International Conference on Mechanical Engineering and Technology (ICMET 2011)* (p. urmeaza a fi publicat). Londra: ASME Press.
78. Pîrvu, B., & Bondrea, I. (2011). Prerequisites for developing a new simulation and manufacturing architecture. *BULETINUL I.P.Iasi - Machine Manufacturing Series, LVII (LXI)* (4), 203-211.
79. Pîrvu, B., Bondrea, I., & Marin, R. (2011). DIGITAL FACTORY: JUST ANOTHER CONCEPT OR A FUTURE INDUSTRIAL REALITY? *The 15th International Conference Modern Technologies, Quality and Innovation (Modtech)*, *II*, pp. 865-869. Chişinău.
80. Pîrvu, B., Bondrea, I., Simion, C., & Marin, R. (2010). Modeling and Control of an Automated Module using Discrete Event Simulation and Object-Based Modeling. (P. Berce, Ed.) *Academic Journal of Manufacturing Engineering , 8* (2), 63-69.Timişoara
81. Pîrvu, B., Schlick, J., & Bondrea, I. (2011). Middleware Comparison: CORBA, COM / DCOM and OPC UA Regarding Procces Orientation. *The 5 th International Conference on Manufacturing Science and Education – MSE 2011*, (pp. 137-140). Sibiu.
82. Pritschow, G. (1997). Trendwende in der Steuerungstechnik. *Konferenz zur "Trendwende in der Steuerungstechnik".* Hannover: Deutsche Messe AG.
83. Pritschow, G., Wurst, K.-H., Kircher, C., & Seyfarth, M. (2009). Control of Reconfigurable Machine Tools. În H. A. ElMaraghy, *Changeable and Reconfigurable Manufacturing Systems* (pg. 78-79). London: Springer.
84. Pritschow, G., Wurst, K.-H., Seyfarth, M., & Bürger, T. (2003). Requirements for Controllers in Reconfigurable Machining Systems. *CIRP 2nd International Conference on Reconfigurable Manufacturing.* Ann Arbor.

85. Ramamoorthy, C., & Ho, G. (1980). Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Trans. on Software Eng. , 6* (5), pg. 440-449.
86. Reinhart, G., Grunwald, S., & Rick, F. (1999). Virtuelle Produktion – Technologie für die Zukunft. *VDI-Z Special C-Techniken* , 26-29.
87. Schleipen, M., & Drath, R. (2009). Three-view-concept for modeling process or manufacturing plants with AutomationML . *14th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)* , (pp. 1-4). Mallorca.
88. Schleipen, M., Sauer, O., Friess, N., Braun, L., & Shakerian, K. (2009). Production Monitoring and Control Systems within the Digital Factory. In G. Q. Huang, K. Mak, & P. G. Maropoulos (Ed.), *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology* (pp. 711-724). Hong Kong: Springer.
89. Schlögl, W. (2007). The way to the intelligent factory. *Fachkongress Digitale Fabrik.* München: Siemens Ag.
90. Schraft, R. D. (2003). Von der Vision zur Realität durch die Digitale Fabrik. *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZwF) , 98*, p. 269.
91. Schuh, G., Klocke, F., Straube, A. M., Ripp, S., & Hollreiser, J. (2002). Integration als Grundlage der digitalen Fabrikplanung. *VDI-Z , 144*, 48-51.
92. Schuh, G., Millarg, K., & Göransson, A. (1998). *Virtuelle Fabrik – Neue Marktchancen durch dynamische Netzwerke.* München: Hanser.
93. Seemann, J., & Wolff von Gudenberg, J. (2006). *Software-Entwurf mit UML 2.* Berlin: Springer.
94. Seiffert, U., & Rainer, G. (2008). *Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz.* Wiesbaden: Vieweg+Teubner.
95. Suh, S.-H., Kang, S.-K., Chung, D.-H., & Stroud, I. (2008). *Theory and Design of CNC.* London: Springer.
96. Sunder, C., Zoitl, A., Christensen, J., Vyatkin, V., Brennan, R., Valentini, A., et al. (2006). Usability and Interoperability of IEC 61499 based distributed automation systems. In IEEE (Ed.), *2006 IEEE International Conference on Industrial Informatics*, (pp. 31-37). Singapore.
97. Takahashi, K., Hasegawa, K., & Miyagi, P. (1999). Hierarchical programming of sequential control systems by mark flow graph. *7th IEEE International Conference on Emerging Technologies and Factory Automation. 2*, pp. 1297-1305. Barcelona: IEEE.
98. Tata, P., & Vyatkin, V. (2009). Proposing a novel IEC61499 Runtime Framework implementing the Cyclic Execution Semantics. In IEEE (Ed.), *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference*, (pp. 416 - 421). Cardiff.
99. Thompson, D., & Watkins, D. (1997). Comparisons between CORBA and DCOM: architectures for distributed computing. *Technology of Object-Oriented Languages. Tool 24*, pp. 278-283. Beijing : IEEE.
100. Thramboulidis, K. (2005). IEC 61499 in Factory Automation. *International Conference on Industrial Electronics, Technology & Automation (CISSE-IETA 05)*, (pp. 1-9).
101. Vyatkin, V. (2007). *IEC 61499 Function Blocks for Embeddedd and Distributed Control Systems Design* (1 ed.). Research Triangle Park, North Carolina, USA: ISA.
102. Vyatkin, V., & Chouinard, J. (2008). On comparisons of the ISaGRAF implemtation of IEC 61499 with FBDK and other implementations. *6th IEEE International Conference on Industrial Informatics* (pp. 289-294). Dajeon: IEEE.
103. Waßmann, R. M. (2010). *Herstellerneutrale Kommunikation in Industrieprozessen mittels OPC Unified Architecture am Beispiel der SmartFactoryKL.* University of Kaiserslautern, Kaiserslautern.
104. Westkämper, E., & Zahn, E. (2009). *Wandlungsfähige Produktionsunternehmen - Das Stuttgarter Unternehmensmodell.* Berlin Heidelberg: Springer.
105. Westkämper, E., Bierschenk, S., & Kuhlmann, T. (2003). Digitale Fabrik – nur was für die Großen? *Werkstattstechnik online , 93*, 22-26.

106. Wiendahl, H.-P. (2002). Auf dem Weg zur „Digitalen Fabrik“. *Werkstattstechnik online , 92*, 121.
107. Wiendahl, H.-P., & Heger, C. (2001). The transformable Factory - Strategies, Methods and Examples. *1st International Conference on Agile, Reconfigurable Manufacturing.* Ann Arbor.
108. Wirfs-Brock, R., Wilkerson, B., & Wiener, L. (1990). *Designing Object Oriented Software.* Englewood Cliffs: Prentice-Hall.
109. Wollert, J. F. (2009). *Vorlesung Bussysteme in der Automatisierungstechnik.* Kaiserslautern.
110. Yamalidou, E., Patsidou, E., & Kantor, J. (1990). Modeling discrete-event dynamical systems for chemical process control – A survey of several new techniques. *Computers & Chemical Engineering , 14* (3), 281-299.
111. Zäh, M. F., & Müller, S. (2004). Referenzmodelle für die Virtuelle Produktion. *Industrie , 20*, 52-55.
112. Zäh, M. F., Schack, R., & Munzert, U. (2005). Digitale Fabrik im Gesamtkontext. *2. internationaler Fachkongress Digitale Fabrik in der Automobilindustrie.* Ludwigsburg.
113. Zäh, M., Fusch, T., & Patron, C. (2003). Die Digitale Fabrik - Definition und Handlungsfelder. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* , 75-77.
114. Zhang, J., Chang, C., Chung, J.-Y., & Kim, S. (2004). WS-Net: A Petri-net Based Specification Model for Web Services. *IEEE International Conference on Web Services* (pp. 420 - 427). San Diego: IEEE.
115. Zhou, M. a. (1992). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.* Boston: Kluwer.
116. Zühlke, D. (2009). *Systementwurf und -modellierung.* Kaiserslautern: pak.
117. Zühlke, D. (2010). Towards a Factory of Things. *Annual Reviews in Control , 34* (1), 129-138.
118. Zühlke, D. (2009). *Vorlesung Werkzeugmaschinen II - Steuerungstechnik.* Kaiserslautern: pak.
119. Zurawski, R., & Zhou, M. (1994). Petri nets and industrial applications: A tutorial. *IEEE Trans. on Ind. Electronics , 41* (6), 567-583.
120. ***4DIAC. (2008, Septembrie). Preluat de pe Framework for Distributed Industrial Automation and Control: http://www.fordiac.org/27.0.html
121. ***Bosch Rexroth AG. (2009). Bosch Rexroth AG. Preluat pe Decembrie 5, 2009, de pe Bosch Rexroth AG: www.boschrexroth.com
122. ***Brockschmidt, K. (1996, Mai). Preluat pe Aprilie 4, 2011, de pe MSDN: http://www.microsoft.com/msj/archive/S2EC.aspx
123. ***CORBA/IIOP 2.3.1 Specification. (1999, Octombire 7). Preluat pe Martie 26, 2010, de pe OMG: http://www.omg.org/cgi-bin/doc?formal/99-10-07
124. ***CORFU. (2011, Aprilie). Preluat de pe CORFU Engineering Group: http://seg.ee.upatras.gr/corfu/dev/publications.htm
125. ***HOLOBLOC. (2011, Martie). Preluat de pe HOLOBLOC Resources for the New Generation of Automation and Control: http://www.holobloc.com/doc/fbdk/index.htm
126. ***IEC 61499 Device Model. (2008, Aprilie). Preluat de pe ISaGRAF: http://www.isagraf.com/get/IEC61499_device_model.pdf
127. ***International Electrotechnical Comimission. (2006, Aprilie). DIN EN 61499-4 Function blocks - Part 4 Rules for compliance profiles. *International Standard* . Geneva.
128. ***International Electrotechnical Comission. (2005, Ianuarie). DIN EN 61499-1 Function blocks - Part1 Architecture. Geneva.
129. ***International Electrotechnical Comission. (2008, Mai). IEC 62424: Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools.
130. ***ISA 95. (2010). Preluat pe Iunie 15, 2011, de pe ISA-95.com: http://www.isa-95.com/subpages/technology/technology.php
131. ***ISaGRAF. (2008, Aprilie). Preluat de pe IsaGRAF: http://www.isagraf.com/pages/products/iec61499technotes/IEC61499_resource_model.pdf

132. ***ISaGRAF. (2011, Iunie). Preluat de pe ISaGRAF: http://www.isagraf.com/index.htm
133. ***International Organization for Standardization, (2009, Decembrie). ISO 6983: Automation systems and integration -- Numerical control of machines -- Program format and definitions of address words.
134. ***International Organization for Standardization, (2008, Martie). ISO 14649: Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers.
135. ***Lefter, C. (2002, Iulie). Preluat pe Aprilie 4, 2011, de pe e-mail REPORT: http://er.adrianmoisei.com/ccplusplus/COM-DCOM-ATL.html
136. ***MathWorks. (2003). *StateFlow User's Guide*. Preluat pe August 31, 2010, de pe http://www.mathworks.com/products/stateflow/
137. ***Microsoft. (1997, September 3). Preluat pe Martie 26, 2011, de pe DCOM and CORBA Side by Side,Step by Step, and Layer by Layer: http://research.microsoft.com/en-us/um/people/ymwang/papers/html/dcomncorba/s.html
138. ***Object Management Group. (1997). *UML Specification 1.1.* Preluat pe August 27, 2010, de pe http://www.omg.org/: Object Management Group
139. ***OOONEIDA. (2011). Preluat pe Mai 3, 2011, de pe OOONEIDA: http://www.oooneida.org/
140. ***OPC Foundation. (2011, Ianuarie). Preluat pe Martie 28, 2011, de pe OPC Foundation: http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC
141. ***OPC Foundation. (2011). Retrieved Mai 1, 2011, from OPC Foundation: http://www.opcfoundation.org/Products/ProductDetails.aspx?CM=1&RI=412&CU=7
142. ***OSACA. (1996, Mai). *osaca.org*. Preluat pe Februarie 2, 2010, de pe http://www.osaca.org/_pdf/isw4emo.pdf
143. ***Rosen, M. (2008, April). *Orchestration or Choreography?* Preluat pe May 02, 2011, de pe BPTrends: http://www.bptrends.com/publicationfiles/04-08-COL-BPMandSOA-OrchestrationorChoreography-%200804-Rosen%20v01%20_MR_final.doc.pdf
144. ***Siemens. (2008). *Siemens PLM Software*. Preluat pe Iulie 10, 2010, de pe http://www.plm.automation.siemens.com/en_us/products/tecnomatix/plant_design/plant_simulation.shtml
145. ***SIRENA-ITEA. (2003). Preluat pe Iunie 16, 2011, de pe ITEA SIRENA project: http://www.sirena-itea.org/
146. ***SOCRADES. (2009, Octombrie). Preluat pe Iulie 23, 2011, de pe SOCRADES: http://www.socrades.eu/Home/default.html
147. ***Softing. (2010). Preluat pe Mai 3, 2011, de pe Softing: http://www.softing.com/home/en/index.php?navid=0
148. ***Softing. (2011). Preluat pe Aprilie 5, 2011, de pe Softing: http://www.softing.com/home/en/industrial-automation/products/opc/dcom/index.php
149. ***Swain, J. (2009, 12 30). *Simulation Software Survey.* Preluat pe 01 10, 2010, de pe Lionheart Publishing: http://www.lionhrtpub.com/orms/surveys/Simulation/Simulation.html
150. ***Tecnomatix Plant Simulation. (2008, Iunie 13). Training Plantsimulation8.1. Sibiu, Romania.
151. ***VDI. (2008). *Digitale Fabrik Grundlagen - Blatt 1.* VEREIN DEUTSCHER INGENIEURE. Düsseldorf: VDI.