University of
South Wales
Prifysgol
De Cymru

On Virtual Commissioning of Manufacturing Systems

Proposals for a systematic VC simulation study methodology and a new simulation model building approach

Peter Hoffmann

Faculty of Computing, Engineering and Science

A submission presented in partial fulfilment of the
requirements of the University of South Wales/Prifysgol De Cymru
for the degree of Doctor of Philosophy

This research programme was carried out in collaboration with the
University of Applied Sciences and Arts Hannover, Germany

August 2016

Abstract

The development of manufacturing systems is faced with progressively tightening timeframes, along with growing requirements on planning quality and engineering accuracy. These demands result from significant cost constraints, shortening of product life-cycles, increasing number of product variants and economic needs for rapid time-to-market. Thus, an efficient production ramp-up including the commissioning as the crucial part, becomes more and more important for engineering companies to stay profitable. Virtual Commissioning (VC) is widely considered as promising method to address the challenges associated with real commissioning, but the simulation model building necessary for VC is affiliated with considerable effort and required expertise. VC of manufacturing systems has been a research topic in academia and industry for far more than a decade. Positive results are reported from large companies e.g. from the automotive industry, which are mostly utilising the complex and costly suites of tools in the context of the Digital Factory, rarely from SMEs. However, in particular also SMEs are forced to improve their engineering and commissioning processes, but suites of tools and methodologies used in large companies are not reasonably transferable to SMEs. Rationale for the rare use of VC, besides its general complexity, are a high modelling effort to build the necessary virtual plant models and a lack of availability of methodologies for systematic implementation and reasonable execution of VC.

Thus, the main goal of this research is the development of a new systematic simulation study methodology as general guideline for planning, implementation and execution of VC. It is intended to be notably beneficial for engineers from SMEs, as helpful guideline for planning, implementation and execution of VC and to facilitate the substantially high modelling effort required for VC of manufacturing systems. Besides clarifying the requirements and specifying an environment for VC, the criteria to select an appropriate simulation tool have been established. The proposed modular, component-based simulation model building has been split into specified procedures for “Low-level Component Modelling”, to be conducted for the components of the decomposed real manufacturing system, and subsequent “High-level Plant Modelling” of the virtual manufacturing system. The applicability of these new approaches has been validated by planning, implementing and conducting a VC for a track-bound transportation system with self-driving transport cars on passive tracks, which is the major subsystem of the manufacturing system used as test-bed at the UASA Hannover.

As one main result, a novel workflow for Low-level Component Modelling has been proposed that aims for the gradual relocation of this modelling task as far as possible to the origin of components, in the end the component manufacturers should provide together with the deliverable components their mechatronic component models. This is related to a novel proposal for exchangeable mechatronic component models and an outlined possible implementation with AutomationML.

Acknowledgements

First of all, I would like to express my sincere appreciation to Dr. Talal M. A. Maksoud, the Director of Studies of this research project, for many useful ideas and his enduring support, advice and help within my research work.

I am most grateful to my supervisor team Prof. Dr. Giuliano C. Premier and Prof. Dr.-Ing. Reimar Schumann for their enthusiastic supervision, helpfulness, patience, encouragement, profitable discussions and asking the right questions.

Many thanks are due to all my current and former colleagues at the Automation and Control Engineering Group (UASA Hannover), especially Wolfgang Then and Christoph Göhring for their support during hardware reconstruction of the manufacturing system used as test-bed for research. Grateful thanks as well to Roza Kasterina; she relieved me from many tasks in our busy control laboratory at the final stage of this work. I would also like to thank the bachelor students Henrik Bockstette, Michael Bodmann and Jan Gronau for their support in this project.

I would like to thank the members of staff from the University of Glamorgan, Faculty of Advanced Technology now University of South Wales, Faculty of Computing, Engineering and Science and from the Graduate Research Office for supporting my work – not least financially. Financial support from the University of Applied Sciences and Arts Hannover and the European Union through the Erasmus programme are gratefully acknowledged.

Furthermore, I would like to thank Dr. Roland Wischnewski, Oliver Stern and Sven Severin from RIF e.V. and Dr. Dirk Pensky from Festo Didactic for their abiding support regarding modelling and simulation with CIROS.

My thanks also to Dr. Markus Hoyer, along with Prof. Dr.-Ing. Reimar Schumann, for their initial support when starting this research project.

Finally, I would like to thank particularly my family, especially my parents, for their support and encouragement. Last, but most certainly not least, very special thanks to my wife Dietlind and my son Florian, for their patience, support and understanding of my (not only physical) absences.

Diolch yn fawr!

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AGV	Automated Guided Vehicle
API	Application Programming Interface
AutomationML	Automation Mark-up Language
BPMN	Business Process Model and Notation
BREP (B-Rep)	Boundary Representation
CACSD	Computer Aided Control System Design
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAEX	Computer Aided Engineering Exchange
CAM	Computer Aided Manufacturing
CAPE	Computer Aided Production Engineering
CAPP	Computer Aided Process Planning
CAVE	Cave Automatic Virtual Environment
CE	Concurrent Engineering
CFC	Continuous Function Chart
CIROS	Computer Integrated Robot Simulation
CNC	Computerized Numerical Control
CoDeSys	Controller Development System
COLLADA	Collaborative Design Activity
COM	Component Object Model (Microsoft)
COSIMIR	Cell Oriented Simulation of Industrial Robots (now CIROS)
DCOM	Distributed Component Object Model (Microsoft)
DES	Discrete Event Simulation
DIN	Deutsches Institut für Normung (German Institute for Standardization)
DKE	Deutsche Kommission Elektrotechnik, Elektronik und Informationstechnik in DIN und VDE (German Commission for Electrical, Electronic & Information Technologies of DIN and VDE)
DMU	Digital Mock-Up
EtherCAT	Ethernet for Control Automation Technology
EC	European Commission
ECU	Electronic Control Unit
EIA	Electronic Industries Alliance
EIS	Executive Information System
ERP	Enterprise Resource Planning
FAT	Factory Acceptance Test
FBD	Function Block Diagram (IEC 61131-3)
GMA	VDI/VDE - Gesellschaft Mess- und Automatisierungstechnik (VDI/VDE Society Measurement and Automatic Control)
GUI	Graphical User Interface
HIL	Hardware-in-the-Loop
HMD	Head-mounted Display
HMI	Human Machine Interface
IDE	Integrated Development Environments
IEC	International Electrotechnical Commission
IGES	Initial Graphics Exchange Specification
IL	Instruction List (IEC 61131-3)
I/O	Input/Output
IPC	Industrial Personal Computer
JT	Jupiter Tessellation
LD	Ladder Diagram (IEC 61131-3)
LOD	Level of Detail

MES	Manufacturing Execution System
MIS	Management Information System
MIL	Model-in-the-Loop
MPI	Multi Point Interface (Proprietary Siemens PLC interface)
NURBS	Non-uniform rational B-Splines
OEM	Original Equipment Manufacturer
OLE	Object Linking and Embedding (Microsoft)
OLP	Off-line Programming
OPC	Openness, Productivity and Collaboration (formerly OLE for Process Control)
OTS	Operator Training System
PAC	Programmable Application Controller
PCE	Process Control Engineering
PIL	Processor-in-the-Loop
PLC	Programmable Logic Controller
PROFIBUS	Process Field Bus
PROFINET	Process Field Net
P&ID	Piping and Instrumentation Diagram
QA	Quality Assurance
RFAS	Robot based Flexible Assembly Systems (UASA Hannover)
RIL	Reality-in-the-Loop
SAT	Site Acceptance Test
SCADA	Supervisory Control and Data Acquisition
SCARA	Selective Compliance Assembly Robot Arm
SFC	Sequential Function Chart (IEC 61131-3)
SIL	Software-in-the-Loop
SME	Small and Medium-sized Enterprises
SOP	Start of Production
ST	Structured Text (IEC 61131-3)
STEP	Standard for the Exchange of Product model data (ISO 10303)
STL	Surface Tessellation Language
TÜV	Technischer Überwachungsverein (Technical supervision society)
UASA	University of Applied Sciences and Arts
UML	Unified Modeling Language
VC	Virtual Commissioning
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik (German Association for Electrical, Electronic & Information Technologies)
VDI	Verein Deutscher Ingenieure e.V. (Association of German Engineers)
VDW	Verein Deutscher Werkzeugmaschinenfabriken e.V. (German Association of machine tool builders)
VFAT	Virtual Factory Acceptance Test
VR	Virtual Reality
V&V	Verification and Validation
XML	Extensible Mark-up Language

Company, product, and brand names used in this thesis may be brand names, unregistered or registered trademarks of their respective owners. The use of these brand names and/or trademarks in this thesis is for identification purpose only and does not justify the assumption that rights of third parties do not apply. All used brand names and trademarks are subject without restrictions to country-specific protective provisions and the property rights of their respective owners.

1 Introduction

Manufacturing systems are faced with the demands for cost effective and flexible production in a global competition that is characterised by diversification of production and increasingly shorter innovation cycles.

As result of the growing demand for new, cutting-edge and reasonably priced quality products, the design of manufacturing systems takes place in an industrial environment characterized by significant cost constraints, shortening of product life-cycles, increasing number of product variants and a requirement for rapid time-to-market.

Accordingly, *"modern manufacturing is experiencing a paradigm shift towards more flexibility and reconfigurability (physically and logically) to respond quickly and efficiently to changing production requirements and market demands"* (ElMaraghy et al., 2011). Physical reconfiguration means hardware changes e.g. plant layout or machinery. Logical reconfigurations are software changes, especially major changes of control software frequently caused by the hardware changes. Compliance with these requirements tends to result in more complex manufacturing systems (Lee et al., 2007) making the operation and management more difficult and costly (ElMaraghy et al., 2011).

From shorter product life-cycles it follows that the share of product design and manufacturing system planning, realisation and production ramp-up compared to the whole product life-cycle, becomes more and more important.

Production ramp-up with its growing influence is especially important for the economic success (Reinhart & Wünsch, 2007), and commissioning gains particular importance. Since there is a greatly increasing variety of products and hence many product changes occur, the commissioning also has to be conducted more often (Kiefer et al., 2006, Mandel et al., 2008, Makris et al., 2012). The time spent on commissioning claims a notable share of the whole project duration, spent on the erection of a new production plant (Fig. 1-6). This time has a crucial influence on the total profitability of the products, because with constant product lifetime and shortened ramp-up phase, it is possible to offer the products to the market earlier and thereby realise higher profit, as in the early part of a product's life-cycle, the market prices are normally higher (Wiendahl et al., 2002, Eckes & Wagner, 2006). Besides reduced profit, a delayed delivery of products because of an ineffective production ramp-up can result in lost market shares (Ball et al., 2011). Lee et al. (2007) stress the importance of on time product launches in e.g. automotive industry. The authors state a 32 % reduction of benefit for a 6 months delay for the launch of products with 5 years product life-cycle.

The above considerations lead to a progressively tightening timeframe for manufacturing systems engineering and rising needs for better quality planning and planning accuracy at the same time.

In general, manufacturing systems for discrete parts manufacturing can be arranged in many different configurations. Baldwin and co-authors (Baldwin et al., 2011) identified 44 different types of manufacturing systems from literature and structured them in an hierarchical classification system, starting with conventional system, then line system and cellular system. All the different types of modern manufacturing systems with their increasing complexity have the fact that they are composed of many different components or sub-systems in common, such as:

- Storage / magazines
- Conveyor, handling and transportation systems
- Machining and assembling tools
- Robots with automatic tool-changing systems
- Machine vision systems
- Control and HMI/SCADA-Systems
- Communication networks (Field bus, Ethernet)

These manufacturing systems are often configured as a combination of many off-the-shelf parts and some purpose-built parts or sub-systems. Lee et al. (2007) cited e.g. a commissioning engineer from a German plant manufacturing company with the statement that about 70% of equipment for assembly is standard.

The design of such manufacturing systems is undertaken by a variety of engineers, associated with different departments (conceptual design, mechanical construction, fluidic and electrical engineering, control engineering, HMI design, technical documentation) or even different engineering companies. The planning and engineering requires co-operation between these fractions in the design of a functioning system, which may use a wide variety of discipline specific CAE tools.

Figure 1-1 illustrates the basic principle by using a simplified process with only one plant manufacturer handing over the completed plant to the operating company. The commissioning is executed under the responsibility of the manufacturer and the run-up is conducted under the responsibility of the operating company, whereupon manufacturer and operating company can also be subsidiaries or divisions of the same enterprise. The manufacturer on its part can place orders for single lots on a series of different subcontractors etc.

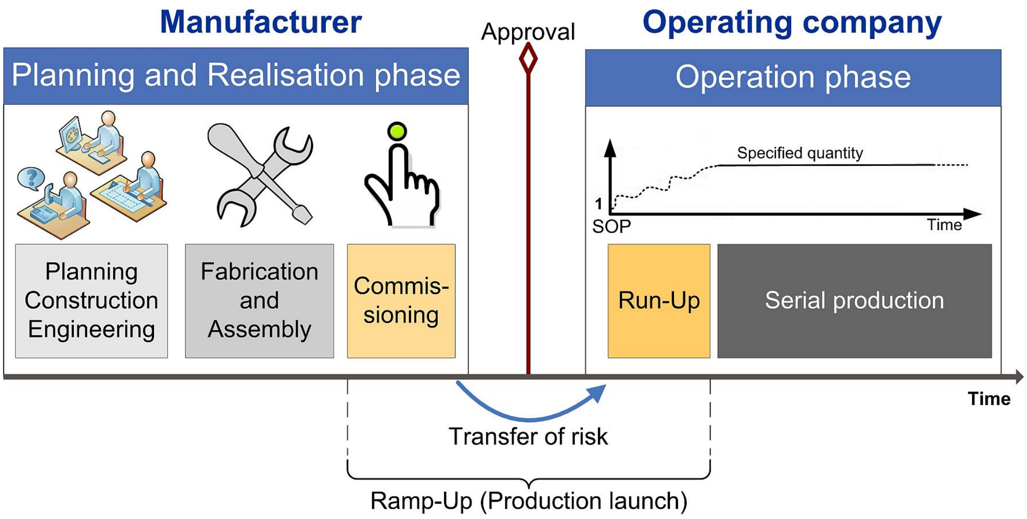


Fig. 1-1: Life-cycle of manufacturing systems from planning to operation - interpreted by reference to (Wiendahl et al., 2002)

Subsequent to the commissioning, is the approval of the manufacturing system by the operating company and thus the transfer of risk from the manufacturer to the operating company as customer (Fig. 1-1).

The life-cycle of a manufacturing system is mostly regarded to begin after the product design and generally comprises several stages, which are often executed sequentially (Haq et al., 2010), with exceptions such as in Concurrent Engineering (CE). CE, also named Simultaneous Engineering, emerged at the end of the 1980s aiming for increased competitiveness by decreasing the lead-time (Sohlenius, 1992). In CE, the phases for design, planning and engineering (see below) are not executed entirely sequentially. Hence, a parallelisation should lead to reduced time-to-market (Bischoff, 2007), even if the required amount of manpower is not necessarily reduced (Addo-Tenkorang, 2011). CE, based on the integration of product design and development of manufacturing process, is able to achieve noteworthy reductions of time-to market, life-cycle costs and engineering change requests as reported in (Pullan et al., 2010). According to Addo-Tenkorang (2011), CE additionally provides better and almost instantaneous communication between the departments involved, providing the opportunity to reduce costly, late changes (cf. Fig. 1-5). Bischoff (2007) uprates this benefit beyond the risk of passing over poorly conceived designs to subsequent departments.

The life-cycle of a manufacturing process starts with the planning and realisation phase (see Fig. 1-1) which covers:

- Conceptual design / plant planning
- Mechanical construction
- Fluidic and electrical engineering
- Automation engineering
- Fabrication and assembly
- Test and commissioning

The following second main phase, the operation phase, includes:

- Run-up
- Serial production

Additionally, the operation phase comprises service and maintenance. A possible, and because of the trend to flexibility and reconfigurability probable retrofitting of the manufacturing system, requires a new commissioning and run-up. The life-cycle of a manufacturing system closes with final shut-down and deconstruction, potentially followed by redistribution and reconstruction of the manufacturing system on a new site.

When the assembly of a manufacturing system is completed, the subsequent ramp-up (production launch) starts with the commissioning focusing on the operational system producing the first completed work pieces able to meet the customer specifications. After approval is received, once commissioning is completed, the operating performance of the manufacturing system is typically not yet optimal or stable and the specified performance parameters (such as specified output quantity measures) will normally be achieved after this point, depending on the success of numerous optimization procedures. Because of this typical ramp-up sequence, it makes sense to split the production launch into commissioning and run-up phase (Zeugträger, 1998, Wiendahl et al., 2002).

The term 'ramp-up' is used inconsistently in literature. A choice of different definitions from literature is given in (Ball et al., 2011), at which *"ramp-up is the period between completion of development and the full capacity utilisation"* is fitting here, whereas other definitions rather describe the phase named run-up in this thesis. The run-up phase (see Fig. 1-1) which follows the start of production (SOP) transfers the operational manufacturing system into serial production conditions, fulfilling the required production quality and the specified output quantities (Reinhart & Wünsch, 2007). This stable and nominal performance is reached through optimisation and stabilisation of the operating behaviour of the process, by solving technical, personal and organisational deficiencies and by the correction of early failures (Zeugträger, 1998).

Besides the failures occurring in earlier phases, another notable part of the imponderabilities during ramp-up is the relatively high risk of component failures during the commissioning process.

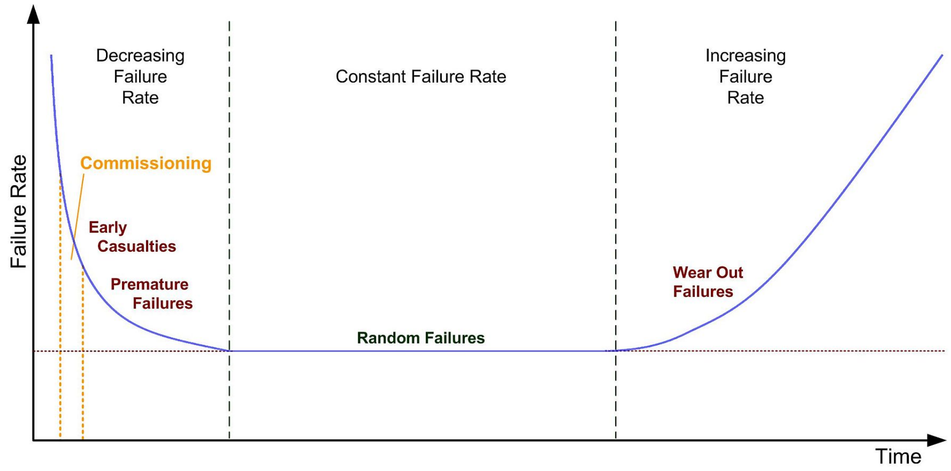


Fig. 1-2: Time dependent failure rate of components – "Bath-tub curve" according to the approximations proposed by Konnov (2007)

The time dependent failure rate of components and systems is well known to be a "bath-tub curve" which derives from its typical shape (Fig. 1-2). The first phase is characterised by a relatively high but decreasing failure rate caused by early casualties in terms of premature failures during origination from e.g. material defects, faulty manufacturing or incorrect dimensioning. Commissioning and run-up are conducted during this phase of relative high failure probability, and it would be astonishing if a system containing up to several thousand new components would not be affected by such early failures (Weber, 2006).

The second phase, characterised by a relatively constant failure rate, caused only by random failures, represents the normal operating conditions and period in the lifecycle for the serial production. An increasing failure rate in the third phase arises from wear related failures caused by deterioration, abrasion and so on. Possible approximations for the three phases (Fig. 1-2) may be represented by a Weibull distribution where different parameters are suggested in (Konnov, 2007).

According to Eckes and Wagner (2006) no functional testing or commissioning of manufacturing systems designed for serial production would generally be conducted after the fabrication and pre-assembly of the system at the manufacturer's site, whereas Haq et al. (2010) report a partial commissioning and verification of new assembly machines by customer witness teams at the manufacturer's site. Such a verification of correct installation and functionality of (sub-) systems at the vendor's facility, is generally called FAT (Factory Acceptance Testing). The correspondence with contractual requirements has to be verified by the examination of the installation according to drawings and specifications as well as a test run (VDI/VDE, 2010). The test run as part of the FAT, conducted either as a traditional function test or by simulation in Virtual FAT (VFAT), shall offer confidence to the end user, that the system *"performs as expected under the full range of foreseeable conditions"* (TÜV, 2014). Third-party companies or service providers such as the German technical supervision society TÜV can also be assigned to execute the FAT. Although the systems are configured in the same way as they will be configured at the plant of the operating company (Dubey, 2011), some aspects can only be verified at the final operation site (VDI/VDE, 2010), especially if only a sub-system goes through the FAT.

Final assembly and commissioning take anyway place at the customer's site, in the facility of the operating company. This on-site commissioning at the operating company has to be regarded as particularly critical, because the commissioning team has only limited access to the resources they may have at the manufacturer's site, especially if commissioning teams are on worldwide duty. The engineers need a wide variety of information about the system functionality and the kinematic and control software sequences for component assemblies or sub-systems in order to conduct troubleshooting and commissioning. These complex ramp-up processes impose challenging demands on the highly skilled interdisciplinary commissioning teams, which are necessary in such circumstances. Every delay and problem which occurs, and the remedial work carried out, particularly where subsequent amendments are required, are of considerable interest and concern to the customer (Eckes & Wagner, 2006). Thereby the risk of damaging the manufacturers' image is always present.

The ramp-up process implies a high physical and psychological strain on the commissioning personnel. The manufacturer's commissioning team as well as the customer's staff where applicable, are under considerable pressure to succeed. They typically face unforeseen problems and very long working days (Weber, 2006). The fixed completion deadline and delays from earlier phases create additional time pressure inasmuch as contract penalties impend. Such penalties are mostly included in contracts to reward a faster than expected ramp-up and to punish late completion (Dougall, 1998).

The use of the term 'commissioning' in technical or scientific literature as well as in practical usage often differs quite significantly, and a generally accepted definition is not available. Sometimes other terms (e.g. start-up) are used synonymously. Furthermore, the tasks that have to be conducted during commissioning depend on the type of plant (process plant, discrete parts manufacturing system). These facts lead to different definitions of the term commissioning. These definitions will be presented below, and an own definition will be derived out of them, which is related to the definition of Virtual Commissioning in this thesis (see page 16).

Zeugträger defines commissioning as establishing operational capability of components or subsystems assembled in advance, including testing of single functions and testing of functional cooperation in conjunction with the wider process. The result is a functional system ready for approval. According to Wiendahl and co-authors, the commissioning is the finalisation of the planning and realisation phase during the life-cycle of a manufacturing system (Fig. 1-1).

An early definition is to be found in the German standard DIN 32541 (DIN, 1977), wherein commissioning is only defined as providing a machine or other technical equipment for use. DIN 19246 (DIN, 1991) comprises the process and defines commissioning as appropriate turn-on procedure of the system in conjunction with the process. The European directive on machinery (EC, 2006) defines commissioning (called "putting into service"), as "the first use, for its intended purpose, in the Community, of machinery covered by this Directive".

In literature, other definitions for the term commissioning can be found. (Wenk, 2008) defines commissioning as the completion of the product creation process during which all subsystems are joined together and connected to complete the system. According to (Eversheim et al., 1990) commissioning means to bring about the operational readiness of assembled products in due time, with subsequent verification of the functionality. Eversheim et al. include all working activities necessary for starting and setting up the correct functioning of assembly groups, machines or complex systems previously mounted. The inspection of the appropriate condition and reliability of single components as well as checking for correct assembly of those components, are not considered to be part of commissioning by Eversheim and co-workers, but are ascribed to quality assurance (QA) instead. For these non-functional testing tasks belonging to QA, "pre-commissioning" is a frequently-used term. Typical tasks include adjustments, cold alignment checks or testing the wiring for integrity and continuity.

The verification of manufacturing systems at a customer's site during commissioning can be subdivided into different phases, occasionally termed SAT and SIT. The Site Acceptance Test (SAT) is applied to (sub-) systems with the focus on verification of installation and documentation and the operation of single systems according to design specifications. This testing, based on the results of a completed FAT, if applicable, is performed to ensure that no changes or damages to the systems have resulted from e.g. transportation or reassembly on site. If a functional test has not been conducted in advance during FAT, or if not all aspects could be tested at the manufacturer's site, the missing functional verification tasks have to be carried out at this point. The Site Integration Test (SIT) is conducted after all (sub-) systems have been connected and linked up with HMI/SCADA systems. This testing has to prove the correct functionality of installed systems integrated in the entire plant including data communication. Generally, for a system to undergo a SIT must first have gone through a FAT and/or SAT testing. A definition for the terms FAT, SAT and SIT from the context of process industry is given in DIN EN 62381 (DIN, 2013). Such a differentiation between SAT and SIT is not always evident in technical or scientific literature. In several publications (Allan & Skibo, 2005, Dubey, 2011) for example, only the term SAT is used for the verification procedure at the customer's site.

In contrast to Zeugträger (1998) and Wiendahl et al. (2002), Weber (2006) does not split the ramp-up into commissioning and run-up, probably because of his process engineering background. Weber uses the term commissioning for the whole ramp-up process, and points out its specific and difficult role as the final phase of the realisation and first phase of operation at the same time. This author amongst others specifies the following duties and aims of the commissioning:

- Establishing continuous system operation as specified in the contract (the principal duty)
- Proof of operational safety, reliability and availability
- Remediation of deficiencies and faults from earlier phases
- Training of operating staff

Zeugträger (1998) also mentions the completion of system documentation and adjustment/optimisation of operating parameters as aim of the commissioning.

It is generally accepted that the commissioning process has to be conducted within the shortest possible time with costs as low as possible.

According to an experienced commissioning engineer (Matley, 1969) cited by Weber (2006) the actual goal of the commissioning team is to "get the money back to the bank" and (Kiefer, 2007) is citing Matley's conclusion that an irreparable loss of money occurs if the ramp-up is delayed, because it will not be possible to earn the lost money later.

The following definition of real commissioning will be used in this thesis:

> *Commissioning means to bring about the operational safety and capability of components or sub-systems joined together and connected to the complete manufacturing system, with subsequent verification of the functionality. The testing of control programs for single functions or sub-systems and the testing of control and HMI/SCADA systems for functional cooperation in conjunction with the process are included. The remediation of deficiencies and faults from earlier phases and the adjustment/optimisation of operating parameters and control programs are also part of this procedure.*

After the product design and plant planning phase the following engineering of manufacturing systems includes the mostly sequentially executed phases: mechanical engineering, hydraulical, pneumatical and electrical engineering and finally the automation engineering with programming of robots, PLCs and HMI (Fig. 1-3).

There are many different powerful and specialized CAE tools for design, planning and engineering, often with integrated simulation. However, there are considerable problems regarding data exchange between the different engineering phases and the different associated tools.

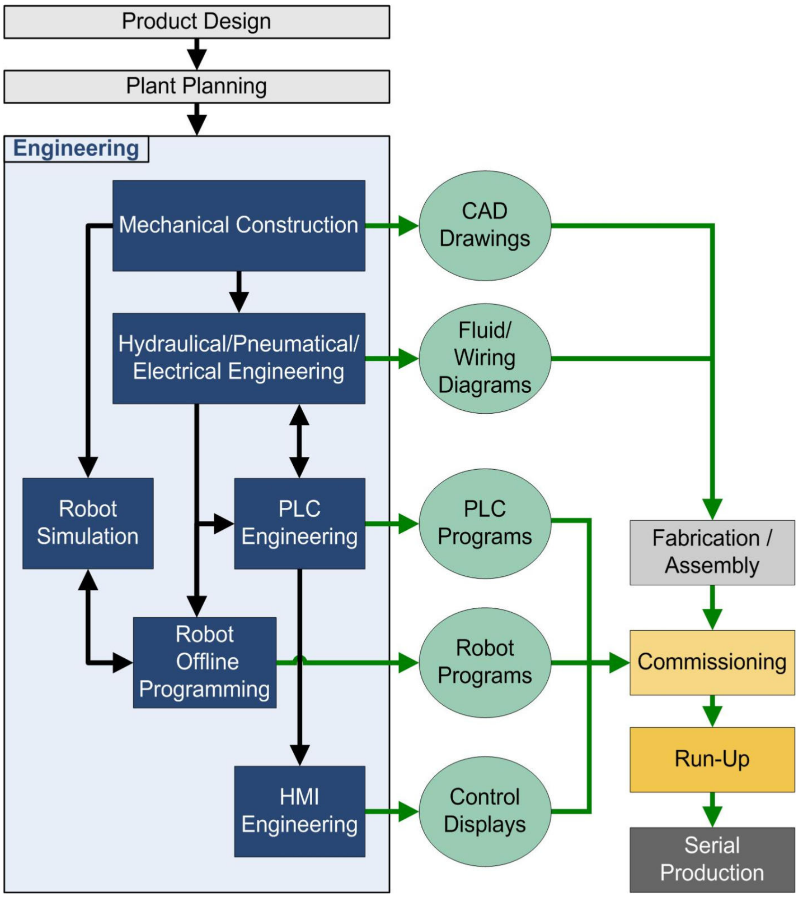


Fig. 1-3: Simplified workflow for engineering and building of manufacturing systems

A typical problem is the need for repeated data entry, often generating random errors. One major problem is the lack of a generally accepted data exchange format (Hoffmann et al., 2009). This might be solved by approaches such as exist in AutomationML.

The outcome of the engineering process is a large number of different CAE documents including CAD drawings, pneumatic/hydraulic plans, wiring diagrams, I/O lists, programs for robot controller and PLCs and control displays for touch panel or PC based HMI systems and similar elements.

Hitherto, after completion of engineering, procurement, fabrication and assembly, the commissioning is finally done. Typically, the validation cannot be conducted before all mechanical, electronics and control systems have been integrated to the entire manufacturing system (Lee et al., 2007). This traditional way of testing, using the real plant and the real control system is still very common. After having eliminated all errors the normal operating phase starts; including service, maintenance and possibly reconstruction.

As conventionally seen, an integrated test of a planned manufacturing system cannot be done before it has been built. The verification of the manufacturing systems' hardware design in connection with related control programs can be generally only be conducted after systems are built and final deployment of controller programs and HMI operator displays is complete. Consequently, design flaws, wiring mistakes, programming errors etc. in significant quantities remain undetected before the first system start-up. This leads in general to corrective measures being required during commissioning, *"which ultimately causes a longer and more costly ramp-up period"* (Haq et al., 2010).

Besides an elongated ramp-up phase, other risks are a high wastage of product (which can be very expensive depending on the process) or, if not completely wasted, a poor product quality and a damage of the newly installed equipment which is also related to the operator safety (Dougall, 1998).

Due to time pressures and the risk of damage, only rudimental failure scenarios are tested during ramp-up, and unidentified errors result in additional time delays and increased costs during the early production phases.

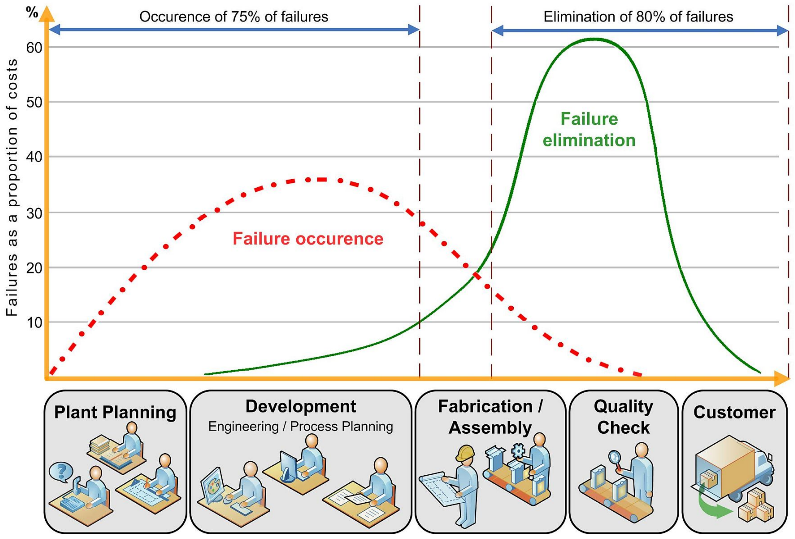


Fig. 1-4: Occurrence and elimination of failures within product life-cycle - based on data of (Pfeifer, 2002)

The term “Quality Check” in figures 1-4/1-5 subsumes all tasks belonging to QA/pre-commissioning (see p. 7) and commissioning. Most of the costly quality deficiencies originate from the planning or development stage, whereas their impact only becomes evident much later (Pfeifer, 2002, Romberg & Haas, 2005). The majority of failures are not detected and eliminated until plant fabrication, assembly or quality checking is undertaken (Fig. 1-4). This means that many failures have to be corrected during commissioning. According to Weber (2006), more than 85% of issues during commissioning are caused by mistakes from earlier phases. The worst-case is not eliminating such failures before handing over the system to the operating company standing for the customer, because this will result in highest possible failure costs (see Fig. 1-5) and a loss of manufacturers’ reputation. Pfeifer refers to several studies, *“that approx. 60 % of the faults which occur in manufacturing have been made before, either in exactly the same form or are similar”*.

The cost of engineering changes rises drastically during the entire development process progressing towards serial production (Pullan et al., 2010). The International Council on Systems Engineering (INCOSE) emphasises the risk of making system design decisions without reliable information from adequate analysis. The published handbook (INCOSE, 2011) stresses the value of preventive risk diminution because it is less expensive to remove errors early in the system development life-cycle (cf. pp. 14/15 of this handbook).

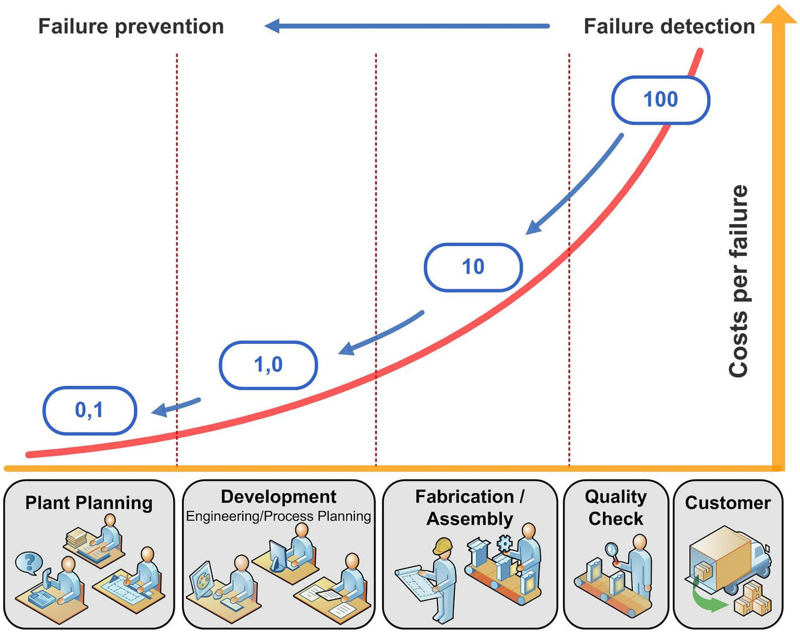


Fig. 1-5: The “Rule of ten” for failure costs

A well known correlation between failure correction costs and failure detection exists, i.e. the “Rule of ten” (Fig. 1-5), which shows exponentially increasing costs per failure by a factor of approx. 10 for each phase from planning to delivery to the customer, if the failure passes undetected (Pfeifer, 2002). According to (Bischoff, 2007) referring to (Burghardt, 2000) the exponential rise depends on project scale, and is between a factor 1:4 for small project and a factor 1:100 for large projects, for the ratio of relative failure costs. Likewise a factor up to 1:100 for the ratio of relative costs to find and fix software errors early rather than later during the software life-cycle, has been stated in (Boehm, 1979a).

The commissioning represents a high and difficult to calculate lead-time and cost factor. Failures from earlier phases are detected all too often only at this stage, and have tardily to be corrected in a time-consuming and cost-intensive (cf. Fig. 1-4 /Fig. 1-5) manner. This fact is not restricted to mechanical construction and electrical engineering; it concerns also particularly the software engineering. Thereby, congregate partially tested control code, possibly inaccurately built systems and at worst, even faulty design.

The accruing tasks are only partly projectable because unexpected failures and troubles often occur. The commissioning is the crucial phase for realizing automated manufacturing systems, because at this time the first test of correct cooperation of all plant sections, sub-systems and components takes place, i.e. the mechanical, electrical, hydraulical/pneumatical and control functions are tested together (Eckes & Wagner, 2006). Even though all components and sub-systems individually work correctly, additional failures often result from their poor cooperation.

Because it is more cost-efficient to avoid failures than to remedy failures, it is better to detect programming errors as early as possible, but even worse are lately detected errors resulting from faulty specified software requirements and design at the beginning of life-cycle. Performing late corrections takes a lot more time and money than would have required in earlier phases. Dougall (1998) concluded that an early test of control code is important *"Experience has proven that it can be up to 20 times faster to find and correct a bug found in the office than in the field after the control software has been installed"*.

Unfortunately, current engineering processes often do not sufficiently reflect an awareness of these facts. Otherwise, control software validation would no longer frequently be the last stage within development process (Spath & Landwehr, 2000), and the testing of e.g. PLC code would be done earlier. A lot of control code is not written before the commissioning phase, but during production ramp-up (Auinger et al., 1999, Reinhart & Wünsch, 2007).

The findings of Eversheim et al. (1990) already indicate that failures of control systems are a main reason for time delays during commissioning. Reasons are the use of new control software not validated beforehand, new communication systems and deficiencies in monitoring and diagnosis

(Reinhart & Wünsch, 2007). Eckes and Wagner (2006) mention an insufficient specification of control tasks in the earlier phases of development and a poor cross-domain information exchange as possible causes. The break in the sequential engineering between mechanical construction, which is done CAD component based, and control software development which is focused on I/O signals, has negative effects right up to the commissioning. As a solution for this *"bottleneck during commissioning"* Haq et al. (2010) propose the development of mechatronic components incorporating mechanics, electrics and hydraulics as well as associated control aspects, which can be stored in a library for re-use after individual pretesting. A mechatronic development of components following this proposal would call for a reduction of breaks between mechanical construction and control engineering. It would also promote a cross-domain information exchange that has been identified by Eckes and Wagner as lacking in sequential engineering.

The validation of complex control software within the development tools for e.g. PLC code fails because of the limited test tools available, consequently the control code is not tested before commissioning (Spath & Landwehr, 2000). A deficient compatibility between sub-systems and the optimisation of complex sequences are responsible for further delays.

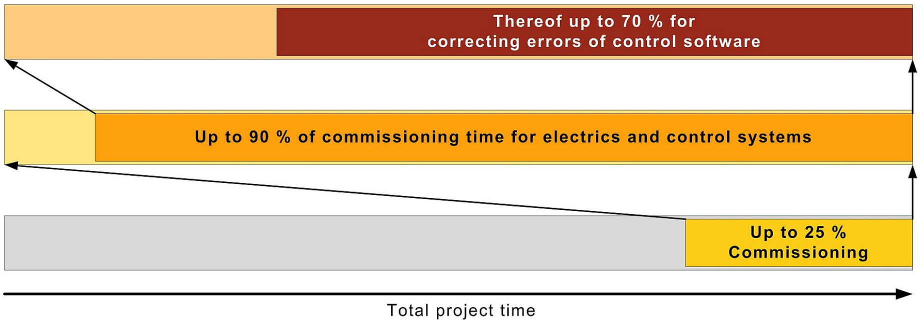

Thereof up to 70 % for
correcting errors of control software
Up to 90 % of commissioning time for electrics and control systems
Up to 25 %
Commissioning
Total project time

Fig. 1-6: Portions of time at the commissioning according to (Zäh & Wünsch, 2005)

In accordance with (Zäh & Wünsch, 2005), referencing a 1997 study of the German Association of machine tool builders (VDW, 1997), the commissioning time consumes up to 25% of the time available for manufacturing system engineering and construction. Most of the commissioning time (90%) is occupied by tasks associated with electrical and control engineering and thereof 70% are caused by errors in the control software. Hence, 60% of commissioning time or 15% of total project time is expended on correcting errors in the control software alone (Fig. 1-6).

The growing demands on manufacturing systems and the functionality of their control software tend to result in software that is more complex. In the past only simple sequences required control, nowadays the control software has to cope with more complex processes and enhanced operating functions, visualisation and handling of disturbances.

Since the 1970s, the historic trend of time dependent portion of engineering domains to system functionality, reflecting development effort and costs, shows a decreasing importance of mechanics (Fig. 1-7). Nowadays complex mechatronic systems with increasing engineering portions and costs arise associated with electronics and more importantly software (Raith & Amman, 1992, Glas, 1993, Wikander & Törngren, 1998, Reinhart & Wünsch, 2007).

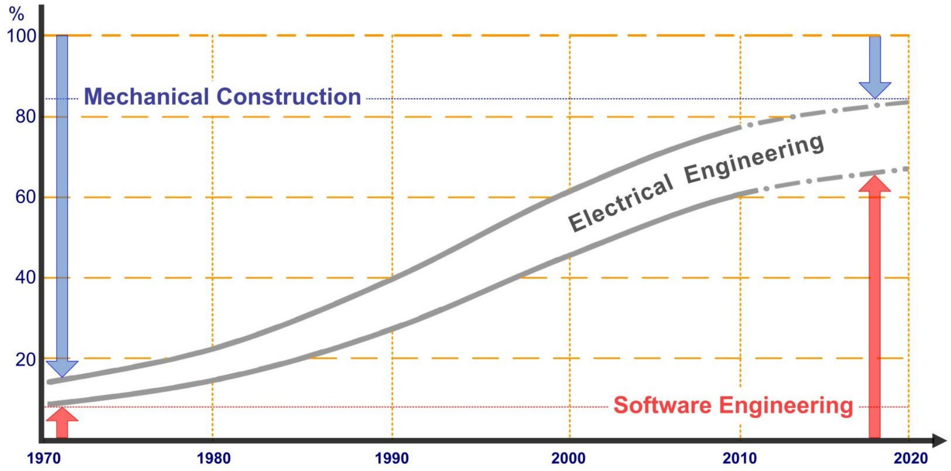


Fig. 1-7: Changing portion of engineering domains contributing to system functionality – based on the estimation of (Reinhart & Wünsch, 2007)

Reinhardt and Wünsch (2007) draw the conclusion that the delays in commissioning caused by errors in control software will tend to ascend proportionally to the growing portion of software engineering (Fig. 1-7) crucial to system functionality which they assume by linking the research results of (Eversheim et al., 1990) and (Glas, 1993).

Due to the trend to more flexible and reconfigurable manufacturing systems, retrofitting of the manufacturing systems requiring re-commissioning has to be conducted more often. Therefore, a challenge arises not only from development of control systems for newly built systems, but also from the necessary modification of control systems with minimal down times during this maintenance phases (Auinger et al., 1999).

The current movement to even more digitalisation in the production (hyped as Industry 4.0), demonstrates the trend towards the increasing importance of software in figure 1-7, as likely to continue. As a result, greater demands on development of control systems and commissioning should be expected.

1.1 Basic idea of Virtual Commissioning

As a possible solution to the above-mentioned problems, several researchers propose Virtual Commissioning (VC) to test manufacturing systems and associated control programs through simulation conducted before the real systems are realised (Hoffmann et al., 2010). During VC, a simulation model of the manufacturing system is used to prepone important tasks of commissioning through simulation, before building the real system. This forwarding of tasks of the real commissioning that are feasible by simulation, is the basic idea behind VC (Fig. 1-8). The prospective goal is the early detection and correction of errors generated during engineering especially in controller programming.

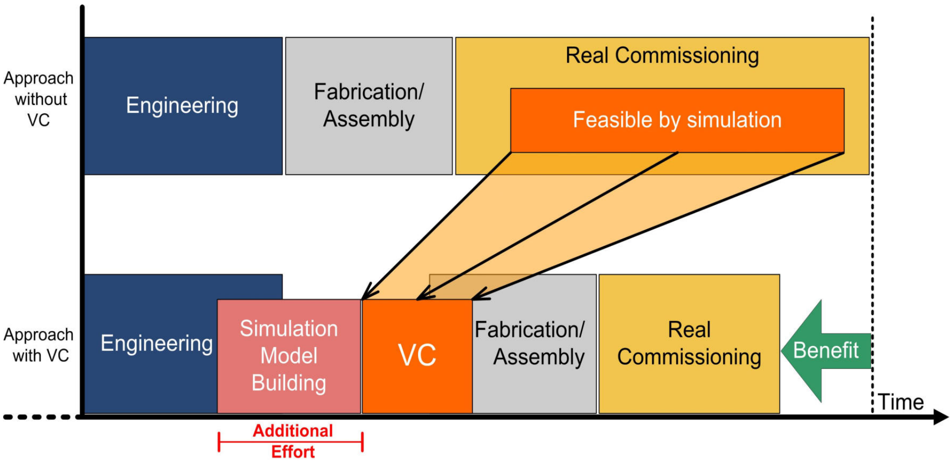


Fig. 1-8: Basic idea of Virtual Commissioning – derived from (Zäh et al., 2006a)

According to (Zäh et al., 2006a, Denkena et al., 2008) an approach with VC shall allow the parallelisation of tasks, instead of a pure sequential workflow in an approach without VC, and thus supports the concept of CE (cf. page 3). A promised benefit, as indicated in figure 1-8, is the time needed for real commissioning and thus the overall time for manufacturing system planning, engineering and erection (lead-time), is reduced. To what extent the total time for the entire commissioning (virtual and real) is reduced, compared to a commissioning without previous VC, depends e.g. on the sort of errors occurring in real commissioning and the scope and detail of VC compared to the real commissioning. It is much more time consuming to remedy detected errors that require even changes of mechanical set-up during real commissioning, than to correct the mechanical setup during simulation or to correct simple programming bugs in control code.

VC allows for testing of various failure and disturbance scenarios that will typically not be tested during real commissioning because of time pressures and the risk of endangering people and equipment. The scope of tested functionalities is therefore not necessarily comparable.

Besides reduced lead-time, conducting a VC promises more reliable planning and thus ensure adherence to delivery dates. In the light of the imponderables associated with commissioning, this is an advantage not to be underrated.

Figure 1-8 is differing from the proposal of Zäh et al. (2006a), in which VC is conducted concurrently with fabrication/assembly, all ending at the same time. It is the opinion of this thesis that the VC should preferably be completed before starting procurement and assembly. On the one hand, this would be advantageous in the case of necessary changes to mechanical set-up identified before assembly or even procurement; on the other hand, it will not be accepted if this results in a large temporal gap between completing engineering and starting fabrication and assembly (Fig. 1-8). However, the expected benefits in reducing debugging and correction efforts during real commissioning and thus reducing lead-time (Fig. 1-8), can only be achieved if adequately detailed mechatronic plant models are available for simulation in sufficient time. The consequence is the necessary minimisation of additional effort and needed time for simulation model building. The faster the simulation model building can be done the earlier the VC can start. It makes sense to start simulation model building concurrently with manufacturing system engineering, but a realistic VC is not possible until detailed engineering design has been completed and the real components have been specified.

Simulation may proceed by starting with product design and plant planning, with e.g. abstract material flow simulation, continuing through the subsequent engineering phases and ending with the realisation of a virtual manufacturing system built of generic or approximate and parsimonious models. Sometimes the term VC is also used to describe such simulations, but the definition used in this thesis is different.

The author intends to use the following definition of Virtual Commissioning in this thesis:

> *VC is the simulation of comprehensively specified manufacturing systems using adequately detailed mechatronic plant models and the preferably original and unmodified control programs and control displays intended for deployment on the real system, which is conducted prior to the building of the real system. The primary function of VC is the detection and remediation of programming errors in control programs and control displays of HMI/SCADA systems, but also of faulty design from either electrical engineering (e.g. misplanned wiring) or mechanical setup, which can be detected too.*

This definition differs from design validation by simulation, typical of early engineering phases. A VC, according to the proposed definition, has the ability to fulfil the aims of real commissioning at a systems level, defined by the author on page 8.

The workflow for engineering and building of manufacturing systems given in figure 1-3, changes with the implementation of such a VC, as depicted below (Fig. 1-9).

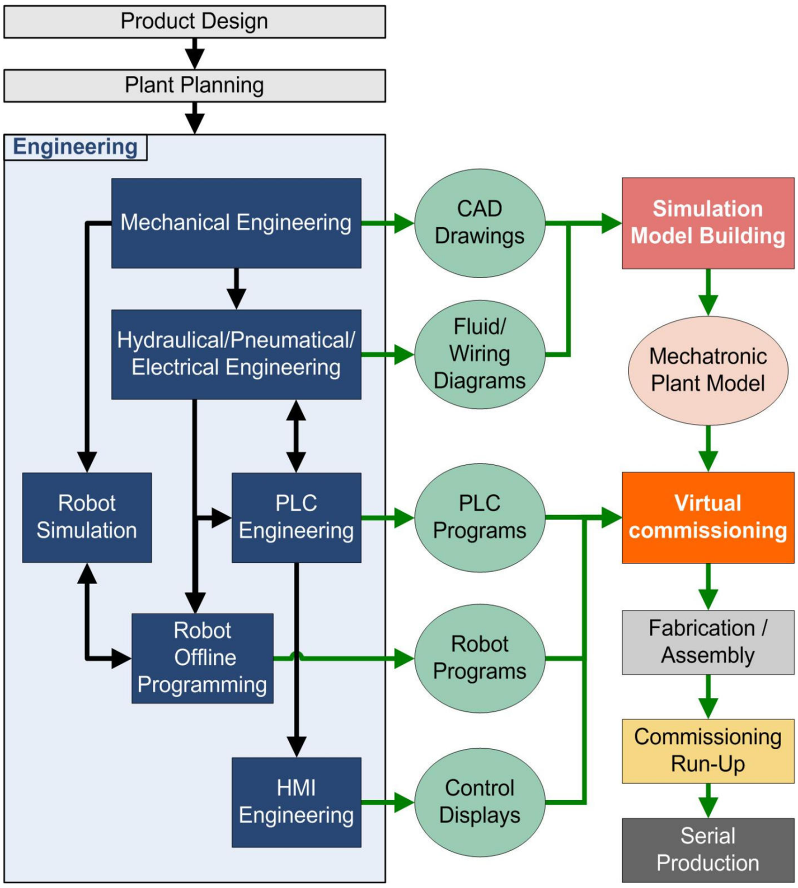


Fig. 1-9: Illustration of simplified workflow for engineering and building of manufacturing systems with implementation of VC

The approach of placing the simulation model building process, combining different engineering data, and the VC (Fig. 1-9) in the workflow between engineering and fabrication/assembly of production facilities proposed in (Hoffmann et al., 2012), has been affirmed and used in (Cruz et al., 2014b). This does not preclude a time overlap of engineering and simulation model building and/or of VC and fabrication/assembly (cf. Fig. 1-8).

Investigations of the promised advantages of VC, such as reduced real commissioning time, better planning reliability and early detection and correction of programming errors, as well as possible drawbacks that have become apparent (such as the modelling effort), will be further developed as part of the literature review. In this review, the particular conditions at small and medium-sized enterprises (SMEs) will be taken into consideration (chapter 2).

Against this background, the underlying hypothesis of this thesis is stated thus:

> *It is possible for SMEs to conduct a Virtual Commissioning prior to full commissioning such that reduced losses would be incurred, than without Virtual Commissioning. Debugging of control programs and control displays would be possible and faults in the planned physical setup of the manufacturing system could also be detected and corrected. This VC approach would reduce fault detection and correction costs and efforts encountered during the commissioning phase of the real manufacturing system.*

The work in this thesis concentrates on the end of the detail engineering design phase, where all design information has been specified by the planning/developing engineers, such that the real manufacturing system can be built.

The Virtual Commissioning methodology for testing manufacturing systems design, associated control programs and control displays, by simulation before they have been built, will be investigated.

The focus of this thesis will be the systematisation of setting-up a simulation environment for VC, suitable for SMEs, that allows testing and debugging of the developed control programs by means of a simulated virtual manufacturing system, before the real manufacturing system is built. The approach is supported by a generalised modelling systematic for all components in the manufacturing system.

1.2 Aims and objectives

For the reasons mentioned above, the motivation of the thesis is to research possibilities for systematic improvement of commissioning and related engineering processes; particularly for reducing the time needed for commissioning and unforeseen failure costs during commissioning.

The new approach should allow the simulation of the manufacturing system, including control and HMI/SCADA systems, before it is built. This should be achieved with only little additional effort for the planning or the commissioning engineer. In order to achieve this, a method must be found to transform the (complete) planning information at the end of the engineering phase into a simulation model as quick and easy as possible, is sought. Using such a virtual simulation of the manufacturing system, the planning or commissioning engineer would be able to detect faults and problems from the simulated virtual plant before the real plant is built, thus reducing the debugging effort during the real commissioning phase and avoiding time consuming and costly rebuilding to correct problems.

A key question for this research project is where to get component models for the manufacturing systems that are to be simulated.

In electrical engineering, typically component manufacturers provide simulation models for chips and other electronic components, often before these are ordered or delivered. In mechanical engineering, simulation models are normally not available or provided by the component manufacturer and there is no widely-used systematic method as exists in electrical engineering, to integrate such data into CAE tools or simulation systems. Nevertheless it would be highly desirable if manufacturers of mechanical or electrical/control components of manufacturing systems would provide simulation models for their components.

1.2.1 The main aim and other goals of the study

The main aim of this research project is to propose a systematic approach for SMEs to facilitate the substantially high modelling effort required for Virtual Commissioning of manufacturing systems. To achieve this, all the specification and design of a simulation model building procedure, which will integrate all available information (e.g. geometry, electrical I/O, kinematic information) into models for Virtual Commissioning, will be investigated.

The complexity of industrial systems is still increasing (cf. chapter 1), hence the depicted problems tend to get worse and particularly engineers in SMEs urgently need guidelines for implementation and reasonable execution of VC. Such guidelines are currently missing nowadays, and one goal of this thesis will be the development and provision of procedural methods applicable for SMEs. Other goals are the clarification of requirements for VC and the investigation of how to reasonably define and set up an environment for VC, which can be confusing due to the many possible configurations and tools.

1.2.2 Specific objectives

To achieve these aims the following objectives have been derived:

- To understand the state of the art in Virtual Commissioning of mechatronic manufacturing systems

  This understanding will be based on the literature review. A VC in this area has to incorporate mechanical engineering, electrical engineering and controller/visualisation programming as well as 3D plant simulation, which is possible in many varying configurations. Thus, plenty of different topics are involved and this leads to a broad scope of required knowledge base.

- To determine the requirements for Virtual Commissioning

  The realisation of a VC project has several technical requirements in form of hardware (e.g. real/virtual controller), software (e.g. tools and their communication), data as well as principal and/or organisational requirements that have to be determined.

- To identify the existing approaches to set up an environment for VC

  Among the variety of existing software tools for simulation and possible coupling to controllers, the approaches suitable for VC have to be identified, and an environment appropriate for this thesis has to be selected and configured.

- To understand the simulation model building and its challenges and how to address them

  The necessary effort for simulation model building is often complained, and this seems to be a reason for restraining, especially SMEs, from implementing VC. It has to be cleared up if, and if so, why this is a problem respectively what are the drawbacks in detail and how they can be addressed. The requisite knowledge for simulation model building has to be investigated in this context too.

- To develop viable procedural methodologies that can facilitate the aims of the study

  These methodologies to be developed have to comprise the implementation and execution of VC in its entirety, having particular regard to the simulation model building procedure.

- To validate the procedures developed in such a way as to indicate the feasibility and desirability of Virtual Commissioning in the engineering environment described.

  This validation will be best done using a real manufacturing system as test-bed. The industry-oriented system at the UASA Hannover has to be reconstructed for VC research, to allow for the implementation and execution of VC according to the proposed methodologies based on this test-bed.

1.3 Structure of thesis

The thesis starts in chapter 1with a broad introduction of manufacturing systems and their planning, realisation and ramp-up to provide the reader with background information of this research. The ramp-up procedure and especially the commissioning as the crucial part of it are introduced in detail.

The presented challenges and deficiencies of commissioning motivate the basic idea of Virtual Commissioning (VC).

Chapter 2 firstly reviews the literature in consideration of mechatronic approaches to improve manufacturing system engineering and available standard testing procedures of control programs. Subsequently, the capabilities of simulation are extensively reviewed. A survey of VC forms an important part regarding existing approaches, its role in the Digital factory and its application in large companies and SMEs.

Chapter 3 presents proposals for a systematic VC simulation study methodology and a new approach for simulation model building. The new proposal for a systematic VC simulation study methodology as general guideline for planning, implementing and conducting a VC for manufacturing systems is intended to be notably beneficial for SMEs unversed in VC, since such a publicly available methodology for VC is currently missing. Even when following a systematic procedure like this, the simulation model building is still affiliated with additional effort. Thus, a new approach for simulation model building based on the preferably unimpeded enlargement of simulation model libraries inside arbitrary simulation tools is proposed.

Chapter 4 describes the building of a test-bed for research on VC system at the UASA Hannover. The manufacturing system existing there is consisting of a transportation system designated as test-system for this thesis and two robot cells. Besides the presentation of its reconstruction for VC research, the system decomposition and modularisation laying the foundation for the simulation model building is clarified. Furthermore, this chapter establishes the understanding how the system works, which is vital for an appropriate modelling. Subsequently, the programming tools (PLC, HMI, robot controller) and 3D CAD tools as parts of the software environment for VC research are specified.

Chapter 5 explicates the requirements for VC (hardware, software and data), including the discussion of some principal and organisational issues, the selection criteria for simulation tools and the simulation model building. The well-founded selection of a suitable 3D plant simulation tool is presented in conjunction with the hypothesis that the novel application of the selected simulation tool to that kind of transportation system at the UASA Hannover will allow for modelling and simulation of this system incorporating PLC and HMI applications. Afterwards, reasons for the specification of the projected VC environment are given. Subsequently, this chapter analyzes the simulation model building process, which is crucial for VC, and constitutes its proposed general splitting into “Low-level Component Modelling” and “High-level Plant Modelling”. Detailed proposals to conduct both tasks following systematic procedures are developed and clarified.

This splitting of model building provides the basis for the proposed novel workflow for Low-level Component Modelling ending in a novel model collection methodology aiming at the enlargement of existing libraries of arbitrary 3D plant simulation tools with mechatronic component models provided by component manufacturers. For this purpose standardised, tool-independent mechatronic component models incorporating at least geometry, kinematics, sensors, electrical I/Os, behaviour/logics and technical data for actuators and sensors are proposed and outlined.

Chapter 6 describes the implementation of VC using the example of the test-bed. This allows for the testing of the new proposals for implementation of VC, especially the simulation model building and its splitting into Low-level Component Modelling and High-level Plant Modelling.

Chapter 7 describes the verification and validation of mechatronic plant model as well as the Virtual Commissioning of the transportation system.

Chapter 8 presents the discussion of results.

Final conclusions from the research in this thesis are drawn in chapter 9 and recommendations for future work are made in chapter 10.

2 Literature Review

In the long lasting cooperation between the University of Glamorgan, now University of South Wales (USW), and the University of Applied Sciences and Arts Hannover (UASA Hannover) several research projects have dealt with the improvement of process engineering regarding modelling and simulation. For example effective process identification (Körner, 1999), design of complex control systems (Syska, 2004) and in (Hoyer et al., 2008) the authors described the way to industrial realisation of the VC for a strictly process plant based perspective, on the simulation model generation from the P&ID in a CAE planning tool. Hoyer et al. demonstrated with the Model^{CAT} approach for bioprocesses, that in principle it is possible to generate simulation models for the VC of such processes in a semi-automatic way, based on the data stored in a CAE planning tool. This allows to rapidly conduct a VC after engineering and programming have been completed, at the latest, because then all necessary data for assembly and real commissioning are available. The basic concepts and related approaches such as operator training systems (OTS) are described in (Schumann, 2007).

For the research presented in this thesis, the focus at the UASA Hannover has switched from bioprocess engineering to manufacturing system engineering, which brings differing challenges e.g. simulation model building for mechatronic components and 3D simulation. The switched focus locates this research project at the interface between mechanical and electrical engineering, controller programming and 3D plant simulation; many different topics are touched and this leads to a broad scope in the literature review. This literature review indicates what has previously been done and conversely showing what is missing and where problems still exist today.

2.1 Approaches to improved manufacturing system engineering

As a solution for the problems depicted in chapter 1 several researchers such as (Kiefer et al., 2006, Haq et al., 2010) propose a mechatronic-oriented engineering of manufacturing systems with development of mechatronic components incorporating mechanics, electrics and hydraulics as well as associated control aspects.

The term “mechatronics” has been formed by Tetsuro Mori from the Japanese company Yaskawa in 1969 (Bishop & Ramasubramanian, 2005), originally only composed for technologies or products incorporating “**mecha**nics” and “elec**tronics**”. This has been extended by other domains, and the definition of mechatronics has evolved since then; a generally accepted definition does not exist today. According to (Bishop & Ramasubramanian, 2005, Thramboulidis, 2008), the definition as *“synergistic integration of mechanical engineering with electronics and intelligent computer control in the* *design and manufacture of products and processes"* (Harashima et al., 1996) is one of the most frequently used.

Thramboulidis states another definition stressing the functionality of mechatronic systems, as these systems *"whose function relies on the integration of electrical and mechanical components coordinated by a control architecture"*. He forms the term "Mechatronic Component", which is composed of mechanical parts, electronics and software, and is characterised by the regulation that it *"cannot and should not be modified before its use; it may only be parameterized for the specific context"*. Furthermore, Thramboulidis claims that the development of mechatronic systems should be based on such components as "basic building blocks" as an aggregation of already defined Mechatronic Components *"without worrying about implementation details of the constituent components"*. Besides advantages such as reusability also possible drawbacks exist e.g. in maybe exponentially rising number of components in real companies, reduced benefit of having already tested components if the need for modification arises or difficulties to integrate available components into systems.

Early approaches for mechatronic engineering and simulation of machine tools and special purpose machinery are focused on software support. During the research project "Föderal" in the context of developing complex mechatronic systems (Föderal, 2001-2004) an approach to designing an integrated engineering platform for mechanical and electrical engineering with programming has been presented (Fehsenfeld, 2003, Angerbauer et al., 2004). The engineering framework, developed by the members of this joint research project, integrates different departments and crafts with the aim to build a construction kit for the creation and re-use of control software, electrical planning and customer documentation (Klemm & Korajda, 2003, Korajda et al., 2004). This approach is related to the change of the sequential engineering process into a mechatronic engineering process (Litto, 2006). A follow-up research project had been "Aquimo" (Aquimo, 2008).

In the same context of developing complex mechatronic systems, the German collaborative research subproject "Mechasoft"(Reinhart & Broy, 2003), a part of FORSOFT II by the Association of Bavarian Research Cooperations, supported by machine tool producers and the University of Munich, presented a model concept for the functional description of machine tools. Using different industrial software tools, partly modified with additional functionality, a hardware-in-the-loop (HIL) simulator for a machine tool has been prototypically implemented (Zäh et al., 2003). This approach to simulation based planning focuses on simulation assistance through all development phases, right from the conceptual design stage, and would provide accompanying simulation and fault recovery of virtual prototypes of mechanical (sub) systems (Anton et al., 2002). Such an approach for mechatronic system design makes sense if the complete design is handled by one company because the models must be developed in parallel to the machine tool design.

The pursued research at the University of Munich in this area dealt with the model-driven development of PLC software for machine tools (Zäh et al., 2005, Zäh & Pörnbacher, 2008), to give developers the opportunity to validate and optimize the control software in the early phases of development. The research aimed at the development of simulation tools that permit verification and commissioning of control software, based on a virtual model of the machine tool. For the modelling of the machine tool a combination of a three-dimensional kinematic model and an executable functional model (sequence diagrams and state machine diagrams) of the machine's hardware components had been used. A similar approach based on mechatronic modules forming a construction kit for production systems in conjunction with virtual prototypes and simulation has been proposed in (Lechler et al., 2008).

Further approaches for the simulation of CNC machine tools are presented e.g. in (Meier & Kreusch, 2000, Potthast, 2002, Suh et al., 2003, Pritschow & Röck, 2004, Abdul Kadir & Xu, 2011, Altintas et al., 2005, Denkena et al., 2008). The use of virtual machine tools for CAM and especially for NC code validation is well established today. However, an integrated simulation that realistically reflects the behaviour of the interacting machine tool, tool and work piece during process (e.g. cutting, grinding) is a topic of ongoing research (Dépincé et al., 2004, Brecher et al., 2009, Aurich et al., 2009, Denkena & Hollmann, 2012). Several industrial tools for CAM/NC code validation are available from e.g. (ICAM, 2014), (Siemens, 2014) and (Dassault, 2014).

Besides the well-established validation of NC programs using a virtual machine tool, Siemens provides an approach for the VC of newly developed machine tools by combination of two Siemens software tools, Mechatronics Concept Designer (MCD) and SIMIT (Siemens-Sinumerik, 2014).

For entire manufacturing systems, Haq et al. (2010) propose a component-based engineering approach. The authors suggest the decomposition of such automated systems into configurable standalone mechatronic components and sub-systems (including mechanics, electrics and fluidics) together with associated control functions that can be tested separately. Lee et al. (2007) illustrated such system decomposition to sub-systems and components for an exemplary manufacturing system. The authors stated a high percentage of standard parts as to be not unusual for this kind of systems. Haq and co-authors recommend furthermore the storage of such pretested components or sub-systems (or their combination to functional groups) for reuse in a library.

2.2 Different approaches for the test of control programs

The test and verification of control programs is possible by applying many different techniques. A comprehensive survey can be found in (Danielsson et al., 2003). The authors defined 18 methods structured in the four main categories: hardware methods, CPU response methods, logic analysis methods and simulation methods and presented their advantages and disadvantages. The following list bases on the four main categories Danielsson et al. proposed, but contains additional annotations.

1. Hardware methods:

- The first method called *"On-line test on the real equipment"* is not a prior test, it describes the real commissioning and its traditional testing with all problems described in chapter 1, or to say it in the words of Dougall (1998) *"a company simply loads the PLC with unproven software, crosses their fingers, and flips the switch"*.

- *"On-line Monitoring"* aims to analyze the control system by means of patch panels, lamps, debug tools, oscilloscope, etc.

- *"Hardwired test panels"*, which allow only a very rough plant simulation by means of electrical components such as transistors, condensers, coils, lamps, switches, thumbwheels, potentiometers, etc. connected to the I/Os of the PLC. Dougall states several drawbacks of this manually operated approximation of real system behaviour such as expensive to implement, difficult to modify, exhausting to use and error-prone. As main disadvantage Dougall stresses that it is almost impossible to test the true interaction between devices by manually forcing. In (VDI, 2015a) this method is called "Signal manipulation" or hardware-based forcing, and it is mentioned that the possibility to test fast sequences or fast processes is limited.

2. CPU response methods:

- Danielsson et al. called this method *"Response program within a single processor"*; Dougall termed it "Emulation within the processor". Both references mean the simultaneous execution of the control program and a response program (simulating the plant behaviour) in the same controller. This response program is easier to change than a hardware test panel, and it does not need so much manual operation because it is easier possible to implement simulated plant behaviour. Danielsson and co-workers state a restricted size of plant models due to the limited amount of RAM memory and CPU power of PLCs. Besides possible timing problems mentioned by Dougall, both references stress as main disadvantage the necessary modification of the control program to communicate with the response program. Thus, because the response program must be removed before real use of the control program, it is impossible to use the verified PLC program on the real PLC without modification. Dougall makes this aware by the statement *"The potential for error at this stage is quite significant and the reality is that untested code is what is being installed on the plant floor"*. This drawback is also stressed in the VDI guideline 3693, additionally it is mentioned that a spatial modelling and a simulation of 3D models is not possible. As advantages the guideline lists: easy to use, no additional simulation tool necessary, complex test cases possible and the modelling language (generally IEC 61131) is known (VDI, 2015a).

- Besides Danielsson et al. mention the possibility to use two or more control systems that are interfaced via I/Os, serial links or networks for this approach

Another method, most likely fitting in this section, is software-based signal manipulation (forcing). Its usage is similar to the hardware-based forcing described above, but the signal manipulation is done inside the controller development software (e.g. manipulating variables in STEP7 online mode) or setting inputs of an emulated controller such as S7-PLCSIM. Like in the hardware based forcing, there is no response program reflecting the plant behaviour, this process knowledge has to be in the analysts' mind to simulate e.g. sensor signals from process.

3. Logic analysis methods:

- Danielsson et al. call the first logic analysis method *"State space search"* as the variation of input variables (up to all possible combinations), either manually or automatic, and the analysis of output state space charts. This analysis is rated to be a difficult task because the state space can be very large. This problem is stressed by other references such as (Park et al., 2006). The authors refer to formal verification approaches of e.g. Thapa et al. (2006), but refute these approaches because of the "state explosion problem". Danielsson et al. stress also that the existing methods to apply rules to automatically detect errors from I/O states have a possible drawback, *"these rules may be as complicated as the logic itself"*.

- The second logic analysis method, termed by Danielsson et al. as *"Lexical analyser"*, implies to transform the control logic to another form or language before analysis.

The research on formal methods such as these generates many approaches; surveys related to formalization of existing PLC code are given by e.g. (Lampérière-Couffin et al., 1999, Bani Younis & Frey, 2003). According to (Frey, 2002), the *"behaviour of the plant is either ignored", "abstracted by some constraints",* which reduces the state explosion problem, or *"completely described by a formal model"* in existing formal approaches. A principal drawback of formal approaches without consideration of plant behaviour is their generally limited verification of PLC programs to the checking of theoretical attributes such as safety conditions, liveness conditions and reachability. Thus this hinders the analysts' determination if the PLC programs achieve the designated control objectives at all (Park & Jang, 2011).

The generation of such plant models is a theoretical and practical problem; this would especially apply for the consideration of the spatial structure of a plant. A formal approach with simple examples (tank, conveyor) is presented in (Lobov et al., 2005), a more complex modular approach in (Machado et al., 2006) and an approach with 2D plant simulation is presented in (Preuße et al., 2011), but a usable industrial implementation is not foreseeable today.

4. Simulation methods: response program is replaced by a simulation model on a PC

Danielsson et al. define six simulation methods:

- The first is termed *"logic simulator method"* and does not make use of the real control system but uses instead a model of the logic, which cannot represent the logic exactly. This approach will be called Model-in-the-Loop (MIL) simulation in this thesis (cf. subsection 2.4.2).

- In the second method, a *"logic emulator"*, connected to a simulation model, provides the advantage that the identical control logic (same programming languages) is used in the emulator as well as in the real controller.

- The third defined method uses *"control system simulators"*. This term is used to describe approaches such as the "Realistic robot simulation" (RRS) where parts of the original control system software, so called robot controller simulation (RCS) modules, provided by robot manufacturers can be integrated into robot simulation systems via the RRS interface (Bernhardt et al., 1995).

- In the fourth simulation method with *"real control systems",* the simulation is connected to the real hardware controller. The principle drawback of this approach is the coupling of a real-time system like a PLC with a simulation model running on a PC, which is mostly lacking in real-time performance and is not synchronised with the PLC. In the case of sensitive timing inside the logic program, this approach tends to show problems. This approach will be called Hardware-in-the-Loop (HIL) simulation in this thesis (cf. subsection 2.4.5).

- If the simulation model and the real system are both connected simultaneously to the control system this fifth simulation method is termed *"on-line control"*.

- The use of a *"control system emulator"*, an exact representation of the real controller, provides principally the advantageous possibility of time synchronisation between simulation and emulator. This sixth simulation method will be called Software-in-the-Loop (SIL) simulation in this thesis (cf. subsection 2.4.3).

Danielsson et al. prefer the last method because the control logic can be tested without modification, it is possible to use the original programming environment for the emulator and the real controller, it is possible to test large systems, it is possible to synchronize simulation model and emulator and because no process hardware or control system hardware is needed in advance. The authors state two drawbacks: only few emulators can be synchronised with a simulation tool and the necessary effort to build simulation models of a production plant – both are to date a problem. For example, the widely used classic S7 emulator PLCSIM cannot be synchronised to simulation because e.g. its time base is not adjustable. This problem has been also identified by Siemens and has been recently addressed by the development of an emulator with virtual time base named SIMATIC S7-PLCSim Advanced (Quirós et al., 2016). The necessary effort for simulation model building is a subject of research in this thesis.

In addition to the single test of control programs described above, a VC according to the definition in this thesis shall be able to detect also programming errors in control displays of HMI/SCADA systems and faulty design from either electrical engineering or mechanical setup. Simulation as the underlying technology is investigated below in detail.

2.3 Simulation

Simulation is widely applied to many different fields such as automotive systems, business analysis, climate prediction, computer systems, electronic circuits, process systems and to manufacturing systems in discrete parts production. In the context of the "Digital Factory", simulation is used for diverse application (cf. subsection 2.7.2). 3D Simulation is the main application for VC of mechatronic manufacturing systems. According to (Shannon, 1976) simulation is *"one of the most powerful analysis tools available to those responsible for the design and/or operation of complex processes or systems",* or quite similar *"one of the best tools available for examining complex system behaviour in dynamic environments"* stated in (Meinert et al., 1999).

The term simulation is defined in various ways in literature. Shannon (1976) defines *"simulation as being experimentation via a model to gain information about a real world system"*. In (Shannon, 1998) the author considers simulation as *"the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system"*. In contrast to Shannon's definition, the design of simulation models will not be regarded as "simulation" but as "simulation model building" in this thesis and only the experimental use of simulation models will be termed “simulation”. Banks (1999) defines simulation as the *“imitation of the operation of a real-world process or system over the time”* and its use to analyse the behaviour of existing or conceptual systems by asking “what if” questions. The VDI guideline 3633 uses the definition *“Simulation is the representation of a system with its dynamic processes in an experimentable model to reach findings which are transferable to reality”* (VDI, 2010).

Generally, it is useful to apply simulation if one or more of the following conditions are present (Shannon, 1976, VDI, 2010):

- Analytical methods do not exist
- Analytical methods are principally available but would require unjustifiable effort or too much simplifying assumptions for model generation
- Complexity of the analysis task (system or process with many influences, dependencies and subprocesses as well as large amount of data, thus tending to be intricate and unmanageable) and need for certainty
- Variable time scale may be required to e.g. accelerate/decelerate the analysis of slow/fast processes
- Simulation may be the only possibility to conduct experiments because of systems’ environment (e.g. space or deep sea)
- Simulation models are reusable

Banks et al. (2010) named several circumstances for which the use of simulation is not appropriate because e.g. a problem can be solved better or more easily by other means. According to these ‘cons’ and to the ‘pros’ listed above, reasons could be:

- The problem can be solved analytically or by common sense
- It is possible and less expensive to perform direct experiments
- The costs will probably exceed the savings
- Necessary resources (personnel, HW/SW) or time are not available and can not be provided
- Verification and validation of models will not be possible
- The management has irrational expectations on simulation, either regarding necessary time for model building and simulation or on informative value (which is related to the necessary modelling effort)
- The system behaviour is too complex or indefinable

Because of the increasing demands on efficiency, quality, and flexibility manufacturing companies are faced with today (cf. Chapter 1), and the increasing complexity of systems and processes (McLean &

Leong, 2001) it has become very important to support the development process of manufacturing systems by simulation (Banks, 1999, VDI, 2010).

According to VDI 3633 (2010), simulation can be used in the main phases of manufacturing systems' life-cycle, namely planning, implementation and operation:

- In the planning phase, simulation is able to improve the design of manufacturing systems. Newly planned systems can be analysed regarding e.g. plant layout, throughput, bottlenecks, sequence logics or control specification. Such simulation is also applicable for the re-design or optimisation of existing systems.
- During implementation, simulation can be useful to analyse the ramp-up behaviour of manufacturing systems by e.g. conducting performance test with different product variants or capacity utilisation. An important application is the test of control software, whereupon not generally original control code is used.
- Typical applications of simulation in the operation phase are e.g. analysis of variants (products, equipment and sequences), control software optimisation regarding scheduling as well as operator training.

Besides this differentiation of simulation use on the timeline, the kind of simulation is also varying on different levels of manufacturing, using different levels of detail (LOD) for simulation models (Table 2-1).

Manufacturing level	Typical kind of simulation	Typical simulation objectives	LOD
Plant	- Discrete event simulation (DES) (e.g. business process simulation and material-flow simulation)	- Logistics and storage - Production principles - Production planning and control	Low
Manufacturing systems / lines	- Material flow simulation - 3D kinematics / physics simulation - **VC**	- System layout / 3D set-up - Material-flow - Control strategies / V&V - System throughput	Inter-mediate - High
Manufacturing cells (incl. robots and machine tools)	- 3D kinematics / physics simulation - Material-flow simulation - **VC**	- Cell layout / 3D set-up - (Offline)-Programming - V&V of control software (PLC, robot and CNC) - Collision test	High
Components	- Finite elements method (FEM) - Multi-body simulation	- Mechanical structure - Non-linear movement diagnosis	Com-plex
Manufacturing methods	- FEM	- Cutting/grinding processes (surface, thermal effects, tool life) - Metal forming processes (filling of form, material flow, tension, strain, fracture)	Very complex

Table 2-1: Typical kinds of simulation on different manufacturing levels – updated table based on (Dépincé et al., 2004)

In a Discrete Event Simulation (DES), the simulation model changes its state only at discrete points in time when an event (e.g. beginning or ending of an activity) occurs (Banks, 1999), the model does not change its state between these events. The intervals between events can be different and the simulation jumps to the next point in time. DES stands in contrast to continuous simulation in which the state of the model is updated in specified fixed simulation time cycles. Continuous simulation with equidistant steps is mandatory for e.g. calculation of trajectories in robotics simulation. Thus, pure DES has only limited use for VC as defined in this thesis.

A plethora of publications addresses DES, and many general findings and advices regarding simulation from authors such as Banks, Carson II., Chwif, Salt, Shannon as well as VDI guideline 3633 referenced in this subsection came from the application of DES (a similar advice for VC is missing) and have been analysed and adapted to VC if applicable. In German-speaking countries, DES associated with simulation on plant level (cf. Table 2-1) is often wrongly restricted to material flow simulation (Wenzel et al., 2010), but in fact already the industrial application goes from system planning and design over implementation and ramp-up to the operational phase. DES is not only used to analyse production and logistics systems (plant planning, engineering and operation) in industry (e.g. automotive, aircraft, shipbuilding and semiconductor) but also in completely different areas. It is also applied in business processes, supply-chain-management, healthcare logistics in hospitals, call centres, logistics (e.g. baggage handling systems, letter/parcel handling, container terminals), transportation (e.g. passenger capacity planning and emergency plans for train stations or aircraft terminals) etc. (Banks et al., 2010, Wenzel et al., 2010).

Typical DES tools are Arena, AutoMod, Delmia (Quest), Demo3D/Sim3D, Enterprise Dynamics (Taylor ED), FlexSim, Plant Simulation (Siemens PLM, former Tecnomatix eM-Plant), ProModel, Simul8, Visual Components (3D Create/3D Simulate), Witness, Xcelgo Experior etc. (Klingstam & Gullander, 1999, Bockstette, 2013, SimPlan, 2016, Wikipedia, 2016). Traditionally, such tools worked with abstract 2D visualisation and did not provide a coupling with e.g. PLCs to run the original control programs (Verl, 2008). Nowadays more and more DES tools provide 3D visualisation and PLC coupling provided as standard can be found too. Approaches to use DES tools for V&V of control code are made since several years (Vorderwinkler et al., 1999, Mueller, 2001a, LeBaron & Jacobsen, 2007, Smith & Cho, 2008, Muller, 2012, Cardoso et al., 2013). Due to the limitation of pure DES for VC (especially in combination with robotics), some vendors of DES software nowadays provide tools for VC in addition e.g. Visual Components with 3D Create/3D Simulate (DES) and 3D Automate (VC) or Emulate3D Inc. with Demo3D/Sim3D (DES) and Emulate3D for VC (McGregor, 2012). The user of Xcelgo Experior for example has to decide by licence if he wants to conduct DES or VC (Bockstette, 2013).

A completely different approach named "Physical Simulation" shows that simulation of manufacturing systems is not necessarily related to simulation software (Cochran et al., 2001). The authors create a "*physical model of the material and information flow within a manufacturing system"* incorporating human operators simulating the workstations and Lego bricks substituting the products. Another kind of simulation uses hardware models of the manufacturing system or plant based on e.g. Fischertechnik (Staudinger, 2015a, Staudinger, 2015b).

2.3.1 Advantages and disadvantages of simulation

Simulation provides many advantages, and the basic idea of simulation is easy to understand. It is easier to justify a simulation study to non-specialists (management, customers) than analytical models. This credibility can be enhanced by validating the behaviour of the simulation model through comparison with the real system (Shannon, 1998). The higher credibility of 3D simulation models compared to analytical models in the context of VC is stressed by (Thapa et al., 2006): *"As per the market situation the customers are not satisfied with the textual verification only, they want to validate the process using virtual commissioning or simulation using 3-D models"*. The crucial advantage of simulation stated by Shannon (1998) and Banks (1999) is the possible test of different aspects (design, layout, operation procedures, material flow, control etc.) of new manufacturing systems prior to implementation or in the case of re-design without disruption of ongoing operations by experimenting with the real system. Additionally, both authors emphasise the better understanding of how a modelled system really works gained through simulation. The possibility to experiment with new or unusual situations and the answer of "what if" questions can be useful for the operator training too.

The most serious drawback of simulation is the necessary simulation model building, because this procedure requires special training to acquire the needed skills (Shannon, 1998, Banks, 1999). Salt (1993) considers *"the ability to keep things simple"* as *"one of the most important of these skills"*. Moreover, the analysis of the system to be simulated and the model building (which are regarded as an art by Shannon and Banks) can be very time consuming and expensive. Meinert et al. (1999) point out that *"the development time to build new models"* but also the effort for *"changes to existing models can be quite substantial"*. The authors suggest modularisation as one possible technique to handle the model building effort in the examined simulation of material handling systems.

The model building is related to the model complexity by the abstraction and simplification processes being necessary for reasonable applicable simulation models. In (Salt, 1993) the author even declares *"Simplification is the essence of simulation"* because amongst others simple simulation models are easier to develop, easier to use, easier to change and easier to understand for the user than complex models.

2.3.2 Simulation model complexity

The simulation model builder is faced with conflictive modelling targets: on the one hand, the model should be as simple as possible to allow easy understanding and manageable computational load; on the other hand, the model should be as accurate as possible. The model of the real system must not oversimplify the system resulting in a trivial or even deceptive model nor contain too many details adding up to a sluggish and expensive model (Shannon, 1976).

According to (Chwif et al., 2000) it is consensus among simulation specialists that a simple model is usually preferable to a complex one. The saying *"Everything should be made as simple as possible, but not simpler"* attributed to A. Einstein, has been paraphrased by the authors to *"A model must be as simple as possible, but not simpler. It must be complicated if necessary, but not so much."* The VDI guideline 3633 (VDI, 2010) postulates that a model *"should not be as detailed as possible, but as detailed as necessary to fulfil the given targets"*.

Chwif et al. (2000) identified "scope" and "level of detail" as crucial factors for simulation model complexity. As reason for simulation models built too complex the authors stated e.g.: the "show off" factor (more impact during management presentation), "include all" syndrome (modeller feels better if all components have been included), "possibility" factor (rising computational power reduces constraints on simulation model building), lack of understanding of real manufacturing system, lack of modelling ability (lack of abstraction – the modeller tries to model as close to reality as possible).

Banks (1999) points out the fact of two models of the same real system, built by two simulation specialists will possibly have similarities, but probably they will not be identical. This is a drawback for the serviceability and reusability of simulation models, because models built by other modellers are more difficult to understand, even more with rising complexity of models. This emphasises the need for model documentation (especially for models provided to others).

2.3.3 Conceptual Modelling

In (Robinson, 2006) the author identifies key aspects of conceptual modelling. In his perception, he includes definition of model requirements and what has to be modelled and how. The conceptual model, a *"simplified representation of the real system",* describes objectives, inputs, outputs, content, assumptions and simplifications of the simulation model to be developed. Banks and Chwif (2010) recommend the creation of a conceptual model prior to the implementation as a computerised representation in the simulation tool, rather than starting from scratch with tool-based modelling. The authors cite a higher modelling effort due to changing conceptualisation, for this recommendation. A generally accepted definition for the term "conceptual model" is missing, but Banks and Chwif look at it as an *"abstraction of the real system being studied"*. They specify the conceptual model consisting of assumptions on system components, structural assumptions defining interactions between these components, parameters and data assumptions. Besides, Banks and Chwif suggest validating if the conceptual model comprises all necessary details to achieve the goal of the simulation study (but not more than this).

2.3.4 Verification and Validation

The terms verification and validation (V&V) have been defined in manifold ways in textbooks, journal articles and standards, and are often used synonymously (Maropoulos & Ceglarek, 2010). Shokry and Hinchey (2009) put V&V in a nutshell by defining its challenge as evaluating if one is *"doing the right things right"*, at which validation assures *"doing the right things"* and verification assures *"doing things right"*. This aphoristic definition goes back to (Boehm, 1979b) and found its way also into the current IEEE Standard for System and Software Verification and Validation (IEEE, 2012).

The validation has to assure that a system is developed, which meets the customer expectations (Andersson & Runeson, 2002) by satisfying its intended use and the user needs in an operational environment (IEEE, 2012). In summary of the majority of definitions given in the comprehensive survey of Maropoulos and Ceglarek (2010) validation focus on assuring that specified requirements base on correct assumptions and are adequate for the intended use, and that this requirements are fulfilled by the developed system. Problems identified during validation can result in a change of requirements specification (Boehm, 1979a). V&V is not only used in software engineering, but also in other engineering domains such as mechanical design. Such a V&V of design is often supported by simulation in virtual environments like in the concept of the "Digital Factory" (cf. subsection 2.7.2).

In the context of simulation model building the validation has to assure that a model correctly represents the real world regarding its intended use (Maropoulos & Ceglarek, 2010), i.e. it solves the right problem by e.g. appropriate system assumptions and correctly modelled physical laws (IEEE, 2012). According to (Carson II, 1989), validation has answer the questions *"How accurately does the model represent reality?"* and if the model can replace the real system for decision making. In short: *"we have built the right model"* (Shannon, 1998). Carson II (1989) states that a model is valid if it is *"sufficiently accurate"* for the purpose of addressing *"what if... questions of interest"* concerning the real system. A similar definition in (Rabe et al., 2008) states that validation *"aims to analyse the suitability of the model related to the given task and the sufficiently accurate modelling of the system under consideration"*. Carson II (1989) who states, *"a model is usually validated to a specific objective"* also stresses this relation to a given task as well as (Sargent, 2008, Law, 2009). As result, a model developed for a special purpose may be only valid for this specified purpose, and not for another.

Boehm (1979a) defines verification as comparison between the basic requirements and successive refinements (e.g. product design, software code) originating from it. In contrast to the validation, problems identified during verification will generally not result in changes of the requirements specification, but in changes of the implementation (e.g. corrections of programming errors). The verification procedure has to assure that the "*system implementation satisfies the validated requirements*" as cited in (Maropoulos & Ceglarek, 2010) being conform with this requirements regarding e.g. for correctness, completeness, consistency and accuracy (IEEE, 2012).

Verification in the context of simulation model building (Fig. 2-1) has to answer the question if a model performs as it should, respectively as the modeller expected and intended (Shannon, 1998, Williams & Ülgen, 2001). Thus, model verification has to ensure if the implementation of the computerised model is correct (Sargent, 2008) and correctly represents the model assumptions (Carson II, 1989). In short: *"we have built the model right"* (Shannon, 1998).

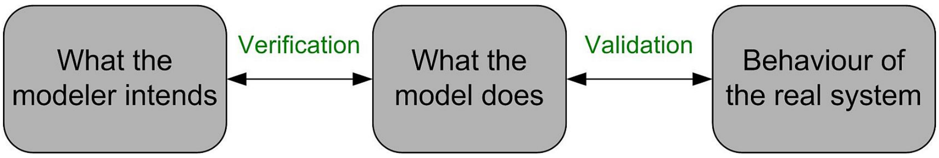


Fig. 2-1: Relationship of simulation model verification and validation (Williams & Ülgen, 2001)

Shannon (1976) stresses the importance of the evaluation (V&V) of simulation models, *"because simulators look real and both modellers and users find them easy to believe"*. Otherwise, *"erroneous results may be accepted with disastrous consequences"*. This view is supported by the recommendation *"Do a lot of verification and validation, not a little"* in (Banks & Chwif, 2011). Because it is not easy to determine errors during V&V of simulation models, Salt (1993) recommends letting domain experts and not simulationists to do this rating.

Gu et al. (2007) observe that the quality of simulation, which means the validity of results, *"highly depends on model accuracy"*. Hence, the authors postulate for the use case of control program validation the necessity of a 100% accurate simulation model, because otherwise *"complex control systems cannot be fully validated"*. The following citations strongly suggest that a request for 100% accuracy is quite unrealistic in this absoluteness but may be applicable for a limited and clearly defined purpose. Rabe et al. (2008) state that verification proves neither the correctness of simulation models, nor the suitability cannot be completely proven. Sargent (2008) remarks that the absolute validation of a simulation model over the entire domain of its intended application would be too costly and time consuming and he rather proposes test and evaluations *"until sufficient confidence"* regarding the model application for its intended purpose has been obtained. Banks & Chwif (2011) even assert *"It's possi-*

ble to invalidate a simulation model, but impossible to validate a simulation model". If a model shows not enough accuracy during validation it is invalid, but a model having sufficient accuracy for different experimental conditions does not prove the validity of the model everywhere in its domain of application (Sargent, 2008). Salt (1993) accentuates that validation may not be possible especially in the case of hypothetical systems; from there Carson II (1989) favours the validation of simulation models by comparison with the real system because it is *"the most objective and scientific method"*. This would indeed require the real manufacturing system being already available during simulation model building. In the case of VC, mostly intended for new systems, this is typically not pre-existing.

The author of this thesis shares the opinion of Rabe et al. (2008) that a 100% model accuracy, as postulated by (Gu et al., 2007), cannot be proven and agrees with Sargent (2008) that a sufficient accuracy for the intended purpose (here the planned VC for a specific manufacturing system) is adequate for simulation model building in an individual case.

A summing up of the simulation model building process including verification and validation can be done using the paradigm presented by Sargent (2008). This paradigm (Fig 2-2) relates the real system with the conceptual model and the simulation model implemented in a software tool with all major tasks relevant to modelling, simulation and V&V.

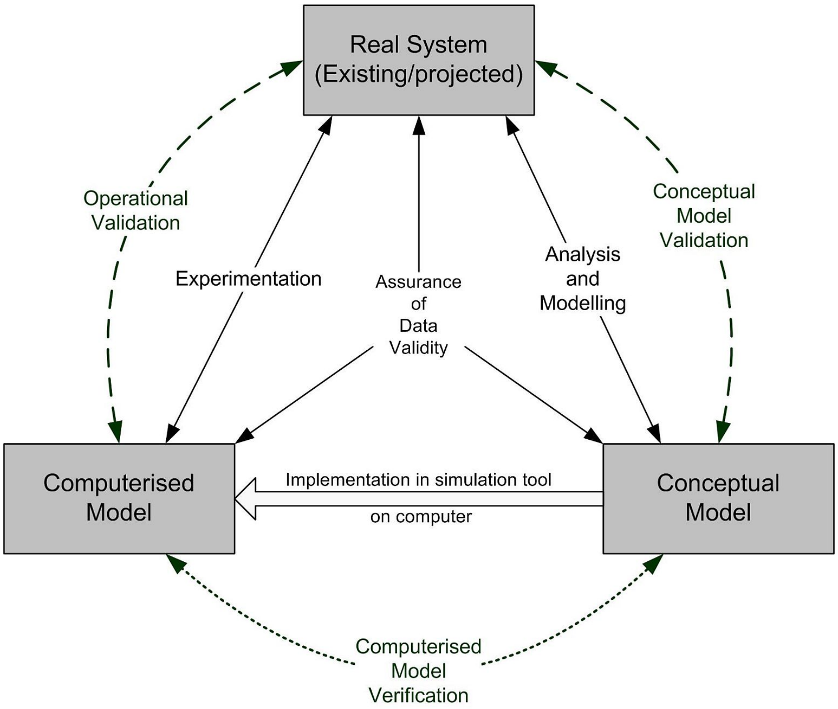


Fig. 2-2: Modelling process with verification and validation according to (Sargent, 2008)

Andersson and Runeson (2002) pointed out the substantial portion of projects budget claimed by the V&V of software systems. The authors reported that code inspection at automation domain has to be more efficient and thus needs improvement.

2.4 V&V of control code by simulation approaches with X-in-the-Loop (XIL)

The so-called X-in-the loop testing methods (in which X stands for model, software, processor or hardware) are often applied in the field of model-based embedded software development and verification for electronic control units (ECUs) in automotive industry as described in (Shokry & Hinchey, 2009).

2.4.1 Model-based development of control software

Model-based development of industrial control with e.g. PLCs is less frequently used in industry. Exemplar research approaches are presented in (Thieme & Hanisch, 2002, Hästbacka et al., 2011, Hoyer et al., 2006) for process technology, in (Lemmer et al., 1995) for a stamping process, in (Vepsäläinen & Kuikka, 2014) for a crane system, in (Zäh et al., 2005, Zäh & Pörnbacher, 2008) for machine tools and in (Brecher et al., 2013b, Andemeskel, 2013, Hossain & Semere, 2013, Barreto, 2014) for manufacturing systems. A survey of several other model-based approaches is given by Vepsäläinen & Kuikka (2014). According to Dubey (2011), only small applications have been used as examples in case studies for model-based control software development and the applicability of modelling frameworks based on UML or e.g. Simulink to industrial scale applications has not been proven.

2.4.2 Model-in-the-Loop (MIL)

MIL, the typical simulation for model-based development, describes the test of control algorithms which are not yet implemented in a standardised PLC language according to (IEC-61131-3, 2003), but in a modelling language e.g. based on finite-state machines, mostly implemented with graphics oriented tools such as e.g. (Matlab)/Simulink. Generally, the controller model is simulated together with the virtual plant (or system) model using the same tool (VDI, 2015a). Co-simulations in which control system and process are simulated within different connected tools are also possible. Vepsäläinen & Kuikka (2014) state that MIL is used to validate the conceptual control solution, but it should not preclude the later uses of e.g. SIL and/or HIL simulations.

2.4.3 Software-in-the-Loop (SIL)

According to VDI 3693 (2015a) the prevailing definition of SIL in publications and industrial use is the simulation of a virtual plant connected to an emulated controller executing real control code (Fig.

2-3), but other definitions can be found in literature. In (VDI, 2011, Machado & Seabra, 2013) for example, SIL is defined as a simulation with real plant and virtual controller, whereas VDI (2011) entitles the simulation with virtual plant and virtual controller "system simulation".

Wischnewski & Freund (2004) emphasise the capability of SIL with an example of a transportation system with more than 3000 I/Os simulated in COSIMIR and connected with a Siemens STEP7 program.

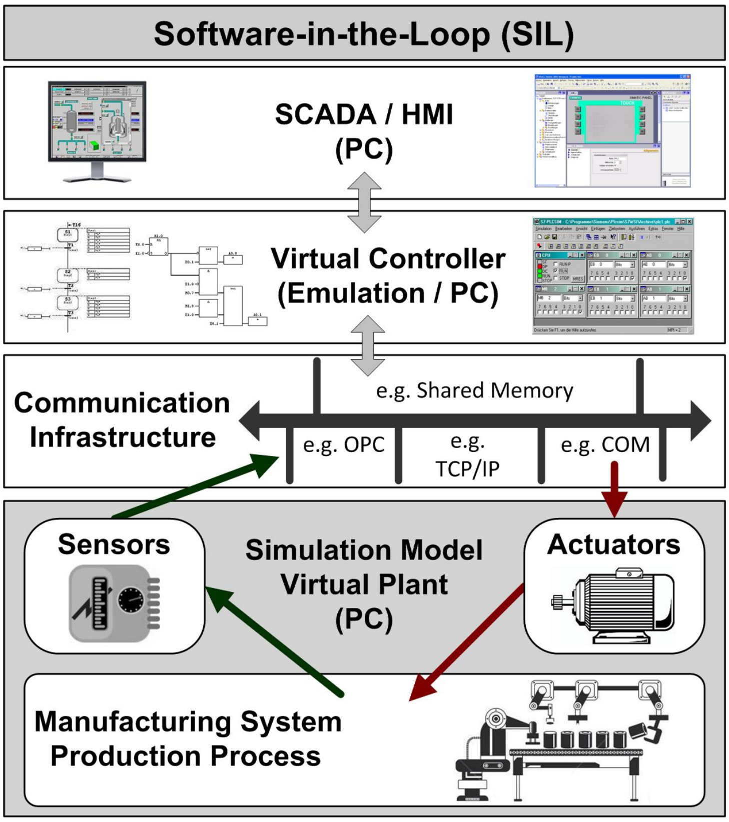


Fig. 2-3: Software-in-the-Loop simulation corresponding with (VDI, 2015a)

Other examples for SIL in manufacturing system simulation are given e.g. in (Montalvo & Phillips, 2010, Li, 2011a, Dzinic & Yao, 2013). Montalvo and Phillips coupled the model of a packaging line in Demo3D/Emulate3D via Rockwell RSLinx Gateway to a virtual controller RSLogix Emulate 5000. Li uses CIROS Studio with its internal virtual S7-controller and Dzinic and Yao established the TCP/IP communication between simulation in Experior and S7-PLCSIM via third-party tool NetToPLCSim using the S7ProSim COM-Interface of PLCSIM. A notably impressive example for SIL from the area of process simulation has been presented in (Krause, 2007). The author established a SIL simulation incorporating a dynamic process model of a liquefied natural gas (LNG) plant with 350.000 global variables and 650.000 internal variables running on a Kongsberg simulator. More than 10.000 I/O values connect the control system to the process. This simulation model has been coupled with an ABB 800xA controller emulator (AC 870P/Melody) via OPC achieving an update rate of 500ms.

The real controller programs can be obtained from automatic code generation e.g. based on the controller model used for MIL, or from manual programming using the editors for PLC languages from integrated development environments (IDEs). The Simulink PLC Coder for instance is able to generate PLC code according to IEC 61131-3 in the form of Structured Text (ST) from Simulink models, stateflow charts or embedded Matlab functions (Mathworks, 2015). The PLC code is either generated in PLCopen XML (Van der Wal, 1999, PLCopen, 2013) or in native file formats of several widely used IDEs such as Siemens STEP7 which do not support PLCopen XML import. Hossain & Semere (2013) point out, that the Simulink PLC Coder supports only a limited number of blocks and not all code from Matlab/Simulink. Another tool supporting automatic IEC 61131-3 compliant code generation from Matlab/Simulink is PLC Link (DEIF, 2015).

An inverted approach, converting PLC programs represented in Instruction List (IL) to a Matlab program, is presented in (Martins et al., 2010, Pereira et al., 2011). This approach is intended to allow a MIL simulation in Matlab/Simulink based on already existing PLC code. A similar approach to convert a PLC program, written in Ladder Diagram (LD) to a statechart model is given e.g. in (Jae Ick et al., 2002).

Automatic PLC code generation is also possible using simulation tools for VC and a plant model, since products such as Delmia Automation V5 (LCM Studio) provide the transfer of internally defined sequential function chart to IEC 61131 SFC (Davidson & Sennö, 2005, Akesson, 2010), with native support of different IDEs. Other examples are Tecnomatix eM-PLC (UGS-Tecnomatix, 2005, Abdul Rahman & Kernbaum, 2007) which had been renamed to Process Simulate Commissioning and is now part of Siemens PLM software, or CIROS Planner, a plug-in for CIROS Studio (RIF, 2015). Both aforementioned tools use a sequence of operations in form of a Gantt chart as input data to generate SFCs according to IEC 61131; Process Simulate Commissioning as native STEP7 (S7-Graph) code, CIROS allows the export to CoDeSys. Gantt charts, going back to (Gantt, 1903, Clark, 1923), are *"useful for displaying schedules, whether produced manually or through some heuristic or optimizing algorithm"* (Wilson, 2003). Gantt charts are to date commonly used in industry to visualise sequences of operations, probably often with MS Excel (Bengtsson et al., 2010), but have some drawbacks. It is intricate to apply Gantt charts to complex, large-scale manufacturing systems, and they *"cannot handle alternatives and arbitrary order"* (Bengtsson et al., 2012). Thus, the use of classic Gantt charts (which neither provide loops) for automatic code generation, tends to be limited.

For this thesis, SIL is considered as a simulation using real control code, executed by an emulated, i.e. virtual controller (e.g. S7-PLCSIM, SIMIT Virtual Controller, IBHsoftec S7-Simulation-PLC, RSLogix Emulate 5000), together with a simulated plant. The control code used in SIL can be obtained from either automatic code generation or manual programming. This definition has been confirmed during the author's participation on the elaboration of the new VDI guideline 3693 (VDI, 2015a) as member of the VDI/VDE-GMA technical committee FA 6.11.

2.4.4 Processor-in-the-Loop (PIL)

PIL is applied to test the real control code on the target processor, typically in form of an off-the-shelf evaluation board, in a non-real-time simulation. The simulation tool automatically conducts different test scenarios to detect e.g. code-generator bugs (Shokry & Hinchey, 2009). Such a test configuration is used for the programming of ECUs in automotive industry but it is unusual for the development of industrial control.

2.4.5 Hardware-in-the-Loop (HIL)

Generally, HIL means a real-time simulation where some of the simulation-loop components are real hardware, and some are simulated. First applications of HIL are documented for flight simulation (Isermann et al., 1999) and missile development (Bacic, 2005). Besides its use in aircraft industry, the prevailing application for HIL up to now is probably software development for the different ECUs in cars (Hanselmann, 1996, Maclay, 1997, Isermann et al., 1999, Bacic, 2005, Shokry & Hinchey, 2009, Taksale et al., 2015).

According to Gu et al. (2007), the control program validation by HIL simulation is not widely utilized in manufacturing automation. The authors name reasons such as: HIL is supposed to be a method for the development of ECUs (probably due to the well-established use in automotive industry) and it seems to generate additional effort and costs for the manufacturing system engineering. Thus, the same concerns in terms of effort and costs many SMEs still have today regarding VC in general (cf. subsection 2.7.4). In (Schetinin et al., 2013) the question why HIL is rarely used in automation engineering is complemented with additional answers: the existing HIL approaches are vendor specific and they cover only specific areas of application, others would require a redesign. Moreover, the authors emphasise the problem of mostly missing plant models and vendor-specific bus connections. Due to shortening development time (cf. chapter 1), many companies are not willing to invest time and money into new approaches like HIL. All these facts apply not only to HIL, but also generally to VC.

According to (VDI, 2015a) HIL is often used for an integration test of the entire automation system, aiming at V&V of completed control programs with the same controller hardware which will be later applied to the plant.

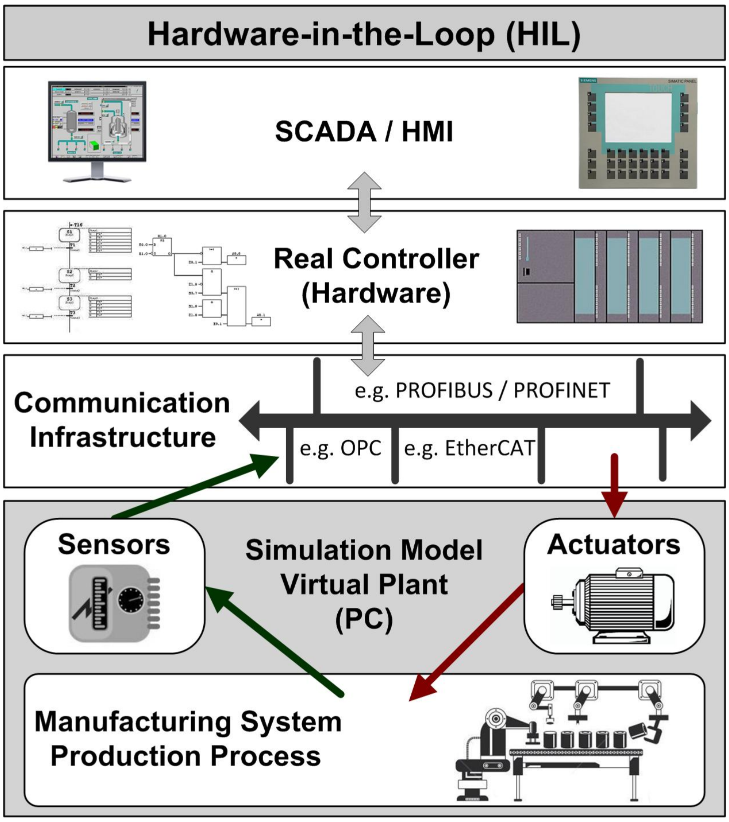

Hardware-in-the-Loop (HIL)
SCADA / HMI
Real Controller (Hardware)
Communication Infrastructure
e.g. PROFIBUS / PROFINET
e.g. OPC
e.g. EtherCAT
Sensors
Simulation Model Virtual Plant (PC)
Actuators
Manufacturing System Production Process

Fig. 2-4: Hardware-in-the-Loop simulation corresponding with (VDI, 2015a)

Members of the VDI/VDE-GMA technical committee FA 6.11 reported that some companies that are aware of the principal advantages of controller program V&V by simulation conduct HIL because they suspect discrepancies between code execution on real controllers and virtual controllers in a SIL simulation. Dzinic & Yao (2013) tested HIL and SIL several times during their case study and the simulation of the production cell with Experior using both methods showed similar results. Such a positive result is not guaranteed for other configurations. Hence, the precaution of some industrial users is occasionally culminating in the use of e.g. PLCs identical to those in the production system with even the same firmware version during HIL simulation.

The coupling between PLC and simulation model can be realised in many different ways. It is possible to support proprietary PLC communication protocols (such as Siemens MPI), Shared Memory, direct coupling via fieldbus or fieldbus emulation by special drivers for the simulation tool, but often HIL is realised using OPC standard (see below). Typical standard fieldbusses for manufacturing systems are often based on serial standard EIA-485/RS-485 such as Profibus-DP, Interbus and Modbus or based on Ethernet (IEEE 802.x) such as Profinet, EtherCAT, EtherNet/IP and Modbus/TCP. Since 2000, fieldbusses are standardised by IEC61158/IEC61784 (HMS, 2016). Some simulation tools allow direct coupling via proprietary PLC communication protocols and standard fieldbusses. A few tools such as Emulate3D and WinMOD support a wide range of PLC communication protocols and fieldbusses (Demo3D, 2016, WinMOD, 2016), others only provide smaller tool dependent choice (Bockstette, 2013). CIROS, currently supporting only OPC for HIL, will support different fieldbusses with the aid of Beckhoff hard- and software (Beckhoff, 2016a, Beckhoff, 2016b) as presented on Hannover Fair 2016.

Fieldbus emulation for Profibus and Profinet is available via Siemens simulation units (Siemens, 2016a) previously referred to as “SIMBA” box. This hardware is currently supported by e.g. SIMIT (Siemens, 2016b), Process Simulate (Popovič et al., 2015) and WinMOD.

An early example for HIL in industrial control is given in (Auinger et al., 1999, Schludermann et al., 2000). The authors coupled the simulation model in Arena with their purpose-built “Soft-Com” interface via TCP/IP with a PLC. Dzinic & Yao (2013) established the TCP/IP communication between a S7-300 PLC and Experior via S7 functions. An example using a standard fieldbus is given in (Erlandsson & Rahaman, 2013). The authors coupled the simulation tool Experior with an Allen-Bradley PLC via EtherNet/IP-CIP. A coupling of controllers and WinMOD via Shared Memory driver (Y200) is presented in (Makris et al., 2012).

For this thesis, the components deemed to be real hardware for HIL simulation are the controller and possibly parts of the real communication infrastructure or fieldbus emulation by hardware units. Thus, HIL is considered as the testing of real control code (obtained from either automatic code generation or manual programming), executed on the real controller, interacting with a simulated manufacturing system via virtual sensors and actuators connected either by real or emulated/simulated communication components (Fig. 2-4).

2.5 OPC

A frequently used technique to establish the connection between controller and virtual plant in SIL/HIL simulation respectively VC is OPC. This client-server based technology *“is a non-proprietary, de facto standard that greatly simplifies communication”* between e.g. PLCs and third party applications, such as HMI/SCADA systems or just now simulation tools, compared to prior necessary specific communication software (McGregor, 2002). Typically, the OPC server either provided by PLC vendor or a third-party tool (e.g. from Softing, Matrikon, Kepware), communicates with the PLC and exchanges the data with connected OPC clients. Brief overviews of OPC and its different classic specification sets, based on Microsoft's Windows technologies' OLE/COM/DCOM such as OPC DA (Data Access) are given in (Hong & Jianhua, 2006, OPC, 2016a) and supplemented with the newer platform independent OPC UA (Unified Architecture) in (Mai & Yi, 2010, OPC, 2016b).

Wischnewski & Freund (2004) reported on the sufficient coupling of small and medium sized simulation models with controllers via OPC, but if it comes to *"many I/O changes"* (not specified), they observed problems during their experiments. In these cases, OPC has not been fast enough to error-free connect PLCs and simulation software running *"in real-time"* (cycle-time not specified). A synchronisation could resolve this, but because this would induce additional programming effort, the authors propose to slow down PLC and simulation in such a case. A deceleration of the PLC is not possible without changes to the program and thus conflicts with the demand to use unmodified programs for VC. Since Krause (2007) demonstrated the successful application of OPC for more than 10000 I/Os (indeed at only 500ms update rate), the problems reported by Wischnewski & Freund (2004) probably arose from poor efficiency of used OPC server.

The use of OPC together with CAPE (Computer Aided Production Engineering) tools has been critically analyzed in (Carlsson et al., 2008) and the authors detected a lack of time synchronisation between simulation and controller system connected via OPC. This results in inaccuracy of robot path. In (Carlsson et al., 2012) the authors specified the problems due to free-wheeling (not synchronised) execution of simulation and controller more detailed. They identified time delays, "Jitter" (random variation of time delays), "Race condition" and "Slow sampling". Race condition describes the error that can occur when two variables change during the same cycle time but are transferred at different times. This would result in different outputs of logic functions. Slow sampling describes the problem of fast signal changes (faster than update rate) being not recognized by the OPC client. Generally, the communication via OPC is not suitable for fast signals, in (Johnstone et al., 2007) the example of an incremental encoder with signal changes every 25 ms is stated because such signals cannot be transferred via OPC with typical update rates of 100 ms.

Barreto (2014) illustrates the influence of the OPC servers update rate. The author uses two Simulink instances on different PCs that are transferring sine wave data via OPC. Running at 100ms, only *"occasional discontinuities in the transmission*" occurred, but an *"evident loss of quality of continuous data transmission"* can be observed, if the OPC server update rate is increased to 500ms, which is too high for the chosen sinus wave frequency.

Many simulation tools such as e.g. Arena, AutoMod, CIROS Studio, Delmia Automation, Demo3D/Emulate3D, Experior, Process Simulate, SIMIT, Tara VRcontrol, Visual Components 3D-

Create/3D-Simulate, Virtual Universe and WinMOD provide an OPC client (Bergert et al., 2009, Carlsson et al., 2012, Bockstette, 2013).

Examples for VC in an OPC based HIL configuration are given e.g. in (Salamon & Heidari, 2012, Guerrero et al., 2014, Hincapié et al., 2014). Salamon & Heidari (2012) use Delmia V6 to simulate a manufacturing cell with 110 I/Os coupled to a Mitsubishi PLC via third-party OPC server. Hincapié et al. (2014) coupled an Allen-Bradley PLCs with Delmia Automation using the Rockwell OPC server. Guerrero et al. (2014) coupled a S7-300 PLC to Process Simulate via Siemens OPC. An example for VC in a SIL configuration from steel making industry, connecting emulated controller, virtual plant model and HMI via OPC is presented in (Kim et al., 2013), The authors reported a response time of below 100ms, rated as sufficient for their specific application.

Concluding, a VC is generally possible by coupling the simulation tool with its OPC client with an OPC server connected to the PLC, but one has to bear in mind the principal problems stated by Carlsson and co-workers. This consideration becomes important when specifying the VC environment for the manufacturing system to be simulated (cf. subsection 5-2).

2.6 Model types for VC

A VC requires the modelling of different processes depending on specific sector of the industry. Due to the considered processes and accordingly implemented systems, it is usual to apply diverse software tools and models. Fig. 2-5 gives a basic hierarchical overview.

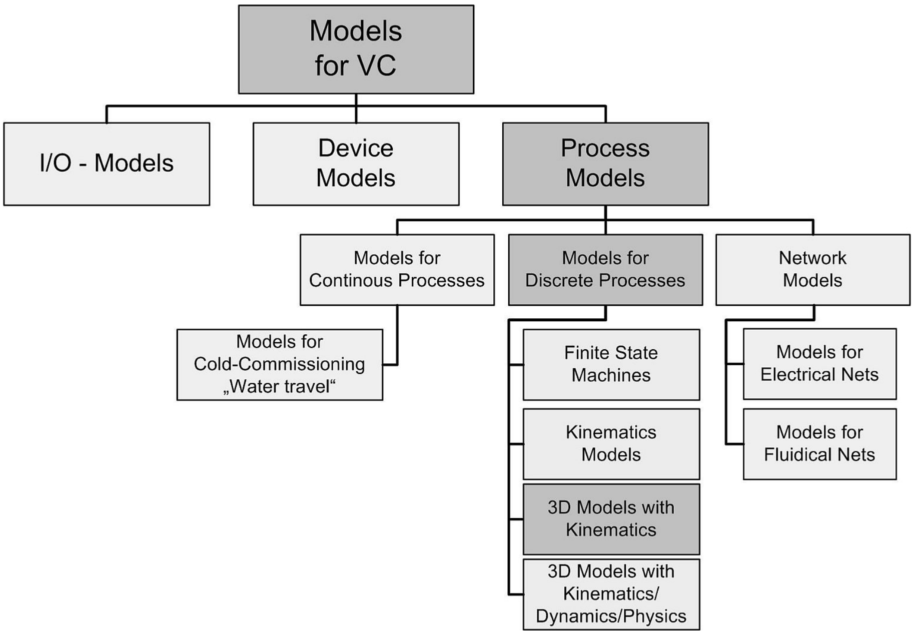

Models for VC
I/O - Models
Device Models
Process Models
Models for Continous Processes
Models for Discrete Processes
Network Models
Models for Cold-Commissioning „Water travel“
Finite State Machines
Kinematics Models
3D Models with Kinematics
3D Models with Kinematics/ Dynamics/Physics
Models for Electrical Nets
Models for Fluidical Nets

Fig. 2-5: Model types for VC corresponding with (VDI, 2015a)

The subject of this thesis is VC of manufacturing systems for discrete processes at which 3D models with kinematics play an important role. Please refer to VDI Guideline 3693 Part 1 (VDI, 2015a) for more details regarding the different other models.

2.7 Approaches to Virtual Commissioning of manufacturing systems

Early approaches to VC, originating from robot simulation, are documented e.g. in (Freund et al., 1994, Jo et al., 1997). These approaches already aim to integrate more than the robot itself into the simulation. Freund et al. propose to integrate e.g. grippers, conveyor-belts, sensors and even automated guided vehicles (AGVs). The approach of Jo et al., called "Virtual testing", is limited to a single robot cell but aims to integrate grippers, sensors and a vision system into the simulation of the entire robot cell.

A broad approach was assumed by the ESPRIT (European Specific Programme for Research and Development in Information Technology) research project "Integrated design, simulation and distributed control of agile modular manufacturing machine systems" (ESPRIT IV 25444 – VIR-ENG). The "VIR-ENG" project has been carried out by De Montfort University, University of Skövde and six industrial partners (amongst others Euromation – a VOLVO subsidiary) and dealt with modular (component-based) machinery design combining control system programming and simulation (Pu & Moore, 1998). Pu and Moore (1998) proposed to divide manufacturing systems into the four subsystems/entities physical machines, actuators, controllers and sensors. The authors postulated that real world systems should be truly reflected by their representation in the virtual world, preferably with seamless integration of virtual and real world. Regarding sensor and controller simulation / emulation, the authors stated the following requirements for a seamless integration:

- Control code which has been developed using the virtual systems can be executed in the real system environment
- The same controllers can be used for the virtual plant model as well as for control of the real manufacturing system
- The sensor signals from virtual plant or real plant can be used equivalent

The integration of PLC programming according to IEC1131-3 standard languages, now (IEC-61131-3, 2003), together with 3D simulation in the so called Modular Machine Design Environment (MMDE) is described in (Adolfsson et al., 2002). The approach was intended to provide off-line programming for automation equipment assembled from components, not only for industrial robots or for CNC machines, allowing the test of PLC control logic before commissioning of real systems. The internal programming language of simulation tools has been only used for describing the uncontrolled component behaviour, for the control of these virtual components, like for real components, IEC standard languages have been used.

Today such a large project might be subsumed under the term “Digital Factory” (see subsection 2.7.2). In the VIR-ENG project, the virtual models of the machine system have been generated by using special tools in a purpose-built framework. The substantive effort for this approach, which is only possible for large companies from e.g. the automotive industry, was justified in (Moore et al., 2003): *“A general solution for designing and off-line programming special purpose manufacturing machinery is not currently available. Such machine systems are typically built from customisable modular automation equipment that is configured from modular components such as sensors, actuators and motion controllers, etc., which are supplied by multiple vendors and typically operate within heterogeneous platforms”*. Such a general solution is still not available nowadays.

Moore, et al., (2003) also referred to the difficulty of 3D simulation model building: *“It should be recognised that the substantial cost of such software packages and the considerable expertise required in building useful models, requires a considered commitment to the use of such tools”*. The simulation model building procedure is still a main drawback which hinders the widely use of 3D simulation for control verification. The building of libraries with mechatronic component models for neither easy re-use nor the import of real components data into the framework have not been in the focus of the VIR-ENG project.

In accordance with Auinger et al. (1999), and due to the possible combinations between reality and simulation, the V&V of industrial control systems can be arranged in four basic system configurations (Fig. 2-6).

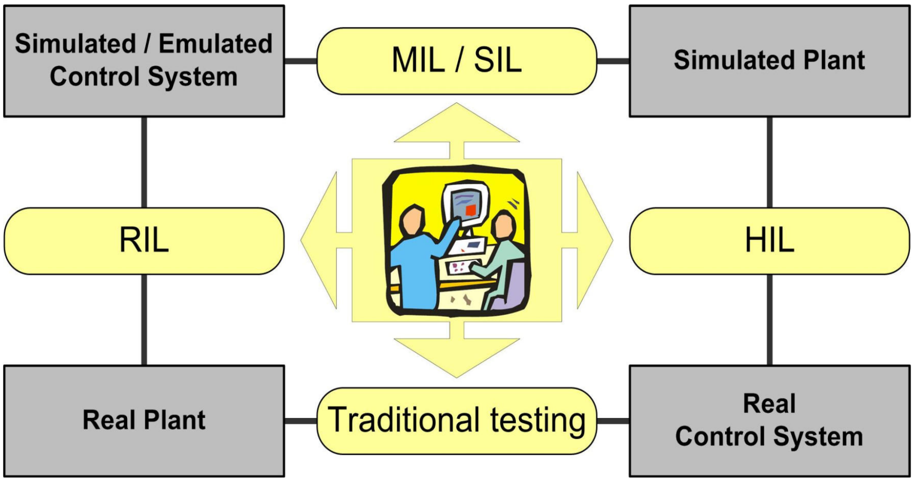


Fig. 2-6: Possible configurations for the verification of control systems

1. Real plant and real control system

This configuration is used for the traditional way of testing and debugging control during real commissioning, with all the problems depicted in chapter 1.

2. Simulated plant and real control system

The combination of simulated plant and real control system is mostly called HIL (see above). An early approach for VC based on this configuration, has been termed as “Soft-commissioning” (Vorderwinkler et al., 1999, Auinger et al., 1999).

3. Real plant and simulated control system

A less common configuration described by Auinger and co-authors, termed “Reality in the loop (RIL)”, with incorporation of real machines into a simulation of the remaining process.

4. Simulated plant and simulated/emulated control system

An entirely software based configuration, termed “Off-line simulation” by Auinger and co-workers is possible in MIL and SIL configuration. In this thesis, MIL is referred to the use of a simulated control system, whereas SIL is seen as implementation of an emulated control system. Another less common term for the configuration simulated plant and simulated control (only execution of pseudo-code, no real PLC programs), used in (Reinhart & Wünsch, 2007, Kong et al., 2012), is “Full Simulation of Machinery” (FSM).

Additionally to these configurations, it is possible to build combined “hybrid” configurations. In (Dumitrascu et al., 2014) the term “Hybrid Simulation” is used for the combination of HIL and SIL. The term “Hybrid Commissioning” is used in (Dominka et al., 2007) to describe the partly, stepwise increasing, integration of a real manufacturing system into a HIL based VC. The concept of “Virtual Fusion” respectively “Hybrid Process Simulation”, likewise based on HIL, has been proposed in (Harrison & Tilbury, 2008, Harrison, 2011, Putman et al., 2015) to integrate virtual models and real components in a modular simulation, where simulation models of robots, machines, conveyors etc. can be replaced by their real counterparts. In (Viswanathan et al., 2011) the authors integrated a virtual robot and a real manufacturing system containing conveyor, controllers and CNC machines in such a “Hybrid Process Simulation”. In (Cardoso et al., 2013) the authors presented the combination of real and virtual subsystems belonging to an educational mechatronic system in a hybrid commissioning scenario.

A VC according to the definition in this thesis (cf. p. 16) is possible in SIL and HIL configuration, not MIL or FSM because the use of original controller code has been requested. Thus, a hybrid version of VC incorporating SIL and HIL for a manufacturing system with different controllers would be compatible with this definition. In contrast, the hybrid configurations combining virtual and real parts of the manufacturing system are not compatible with this definition and thus such use-cases are not a topic in this thesis. For the HIL configuration, the real controller is required in advance, but a VC is realisable before building the plant. The SIL configuration allows a complete VC without any hardware of the manufacturing system being necessary in advance.

Auinger et al. (1999) propose VC because *“Testing and debugging of PLC based control software is still a time consuming and expensive task”*. As a test-bed for their VC approach, the authors used a PLC controlled pallet transfer system with rather simple basic functions but comparatively complex control strategy caused by the overlapping material flow, so that it *“makes sense to fully validate the software before loading it down to the PLC”*. This test-bed has similarity to the test-bed that has been selected for the research project presented in this thesis (cf. chapter 4).

The modelling of the transfer system has been conducted with the discrete event simulation tool ARENA that is usually used for the layout design and throughput optimisation of manufacturing processes. This model in ARENA originally contained all of the control logic (written in simulation tool’s internal control language), and was later reduced to the model used for the Soft-commissioning. Instead of the internal control logic, the reduced model provides nothing more than realistic I/O signals for the coupled PLC and visual feedback. Auinger et al. (1999) found malfunctions of stoppers caused by overlooked inverse logic and problems in the material flow in their experiments. As a result, the authors stated: *“Such failures in the control program would not have been easily detected if using traditional evaluation methods, and in most cases, would appear only after final implementation. Especially for such logical failures, the combination of process simulation and real controller is an optimal method to detect them by a combined logical and logistical check”*. Normally such errors would first be detected during real commissioning. The modelling effort in relation to the benefit was not (probably with good cause) discussed.

McGregor (2002) refers to the benefits of VC (called emulation) and names a shortening of commissioning time by 15-30% and 10-20% more detected errors than with standard control validation methods as stated in (Mueller, 2001b). In (Mueller, 2001a) the author reports of an increased efficiency by 11% more throughput from a bottling line. This became possible by the repositioning of a sensor, which was determined using the VC tool.

An experimental comparative study on Virtual Commissioning conducted with two groups of control programmers, each with 30 individuals, was described in (Zäh et al., 2006b). One group applied VC to the software development for a machine. The authors compared the results to those from the second group of programmers that did not use VC. As test-bed served a tin can moulding press with 27 I/Os controlled by a Siemens S7-300 PLC. One group programmed the PLC and tested the program afterwards in a real world commissioning on the real machine. The other group programmed by using a virtual machine model. They did not execute the real commissioning before achieving successful VC.

The results showed a reduction of real commissioning time by 75%, resulting from enhanced software quality (fulfilment of requirements) at the start of real commissioning. At that time, the authors measured an improvement of software quality from 37% to 84% of correctly realized control functions. The authors stated a resulting reduction of 15% for total time-to-market.

These positive results emphasize the advantages of conducting a VC, but the virtual machine model had already been developed in the run-up to this study, and this effort was not taken into consideration. In their conclusion, Zäh et al., (2006) point out the need for simplified or rather accelerated model building, e.g. by standardization of components or sub-systems to allow a high degree of re-use at model building, which will be a goal of the methodologies proposed in this thesis (cf. chapter 5).

Park and co-workers referred to current approaches for PLC code verification using simulation and Virtual Commissioning. The authors gave an overview of related work and existing tools in this area. Regarding the approach using the simulation tool COSIMIR (cf. subsection 5.2.1), which is also used (now renamed to CIROS) for the research in this thesis, the authors stated *"But modelling and programming in COSIMIR environment are not trivial tasks as it requires thorough understanding of modelling, good programming skills and it takes a long time*" (Park et al., 2006). Park et al. postulated, *"The user need not have deep knowledge of modeling nor should they have good programming skills…"*. The investigation and addressing of such problems is therefore important and will be a goal of the research presented in this thesis. Not only will the PLC code be verified, but also faults in the planned physical setup of the manufacturing system should be detectable.

A III-phase approach for offline verification and validation of control logic is proposed in (Thapa et al., 2006), because *"Till date there are no tools to provide complete integrated solutions for the verification and validation of control logics"*. Thapa and colleagues classify the three phases as: Manual testing, Model checking and Virtual Commissioning. Manual testing means checking the code on a SoftPLC (or emulated PLC like S7-PLCSIM) by user inputs, also known as "Forcing" (see above); this is only reasonably applicable to small programs or program parts. Model checking uses formalization of the existing PLC program as described e.g. in (Bani Younis & Frey, 2003). The standard IEC 61131-3 code is converted to an intermediate language, transformed to timed automata and the model checked. For the VC in the third phase the authors coupled a virtual plant model and an emulated PLC running the verified standard code. The III-phase approach, as other approaches too, requires considerable, costly effort and expertise to build the virtual system for VC.

The simulation tool COSIMIR with its extension module COSIMIR Transport (Wischnewski & Freund, 2004) for carrier based transport systems form the software base for the VC procedure described in (Wischnewski, 2007). The author justifies the VC by the exceedingly error-prone control design of transport systems originating from the use of many sensors and actuators and the complex program sequences required for the routing strategies implemented.

Rossmann et al. (2007) described a VC with COSIMIR using an example of the iCIM manufacturing systems from FESTO. The authors specified a reduced engineering effort by up to 50% and an up to 50% faster start of production (SOP) when all component models were available in the internal simulation model library of COSIMIR. Additionally, a library of reusable controller programs was developed. The additional effort for the first system caused by the necessary simulation model building is specified to have been 30% greater (probably despite of skilled modelling experts being available), with a 20% delayed SOP. This also emphasizes the importance of simulation model libraries with standardised mechatronic components.

Drath and co-workers proposed an "evolutionary" approach for a successful implementation of VC considering industrial requirements (Drath et al., 2008a). They identified the use of real control code and real engineering tools as essential. Moreover they postulated the need for seamless integration of the VC into existing engineering workflows, and *"the additional engineering effort for the creation of the simulation models and the interconnection of PLC and robot programs shall be as low as possible"*. Furthermore, they ask for the support of several PLC systems and the use of virtual controllers that behave identically to the real PLCs. They selected the simulation tool COSIMIR, ABB RobotStudio and a CoDeSys based PLC IDE, to wit ABB ControlBuilder as specified in (Drath et al., 2008b), and coupled virtual robot controller, a soft PLC and the simulation tool via OPC. The virtual I/O planning was done manually with the help of MS Excel. The behaviour modelling of the physical plant, including parameterisation of actuators and sensors, was done manually within the simulation tool, detailed physical behaviour or automation functions need not to be implemented in the simulation model. In (Drath et al., 2008b) the same authors refer to performance of this solution. It has been possible to execute the simulation of a manufacturing cell with four robots using the simulation tool COSIMIR, the soft PLC and a virtual robot controller on one standard dual-core PC. For a complete production line, it would have been necessary to use multiple PCs. In their tests, 50-200 I/O signals per manufacturing cell have been exchanged. The authors regard a cycle time of 100 ms as generally sufficient, only for detailed collision investigations they recommended 10-20 ms. They refer to the problem of very short signals, e.g. changing within 10 ms from '0' to '1' and back to '0', which would not be recognized by simulation (see also problems described by Carlsson). This problem exists also in reality, if e.g. a sensor signal is shorter than the cycle time of a real PLC, in such cases either the signal has to be prolonged or the cycle time of PLC has to be shortened. Besides an increased engineering software quality, the authors expected a middle-term saving potential of circa 10-30% in the real commissioning phase when running a prior VC.

Reduced time and costs of starting or modifying a manufacturing line of up to 50% by conducting emulation (i.e. VC) with the tools considered, i.e. Demo3D and AutoMod, are stated in (Starner & Chessin, 2010), which is assessed by these authors with *“weeks of time devoted to production rather than to building and debugging the line”*. The case study showed that one station on a conveyor system would have been installed in the wrong location; this would have resulted in a malfunction of the system if it had not received prior attention using VC. Starner and Cassin suggest that the misplaced station would have been unnecessarily exchanged for a faster and more expensive sub-system because a solution involving relocation would not have been found in the *“high-pressure situation”* (cf. chapter 1) of real commissioning.

The equipment for the conveyor system in this case study had already been procured and delivered but not assembled. Hence, the VC should be conducted early enough to implement design changes prior to assembly (cf. subsection 1-1). The modelling effort has not been rated, but Starner and Chessin (2010) declare, *“Sometimes the magnitude of the problems discovered may not justify the modeling effort and expense”,* and propose to reconsider the model to justify the effort by adding value, such as its use for operator training. They praise the *“easy-to-use, ready-made catalog items”* of Demo3D/Emulate3D but make no mention of the necessary effort to enlarge the library with new components.

2.7.1 Economic Considerations of VC

Reinhart and Wünsch (2007) investigated the economic considerations of conducting a VC for manufacturing systems. The authors distinguished between the two possible approaches for VC, SIL (termed “full simulation of machinery”) and HIL simulation. As an advantage of the SIL procedure, the authors stated the usability of a virtual time frame to allow the simulation of very complex models on standard PC hardware by calculating the simulation slower than real-time. As disadvantages the authors stated *“low availability of up-to-date control simulation packages for a particular control version”* and the *“rather high abstraction levels of generalised control simulation models”*. The authors postulate that control software can often only be tested with code written in the simulation tool and not the real controller code which avoids an exact reproduction of the control behaviour prevail, and therefore favour HIL. The author of this thesis partly disagrees with the last point, because some simulation tools such as CIROS allow for the internal simulation of PLC code and often it is possible to couple virtual controllers for a SIL simulation, HIL does not give automatically an advantage because then a coupling possibility with real controllers is necessary.

Subsequently the authors classified VC approaches of other researchers that always focus on only one of the following hierarchy level: process level, machine level, cell level or plant level, at which a balancing between modelling effort due to detailing and benefit becomes necessary. The authors' HIL based technical concept is similar to that introduced in Zäh and Wünsch (2005).

Based on their assumption that the attempt to start a VC of a complete production system from scratch would result in too much complexity and effort, and based on their technical concept, the authors proposed a procedure as a guideline for the economic application of VC in four steps.

1. A production system is decomposed into subsystems with defined boundaries and minimal interfaces. Each subsystem is a possible VC project.
2. An evaluation of the subsystems should be done to estimate the efforts and benefits for each subproject.
3. Conducting an effort-benefit-analysis rating all possible VC projects against each other, therefore an economic area and an uneconomic area are defined by drawing a "complexity border" and a "rentability border" where the possible VC projects are arranged.
4. Developing a VC strategy by the analysis of completed old projects and their problems regarding project cycle time, software quality and costs limits. Out of this analysis the choice for a VC subprojects to start with has to be made.

This guideline gave some hints on how to reasonably sub-divide the overall VC challenge of this research project at the UASA Hannover into different tasks.

2.7.2 VC in the context of Digital Factory

The continuation of such projects as "VIR-ENG" (see above), mainly driven by automotive industry, lead to the concept of the "Digital Factory" respectively "Virtual Factory". In literature, both terms are often used synonymously. A differentiation is given by (Westkämper et al., 2009) defining the Digital Factory as database oriented static digital copy containing models of objects and resources of the real factory whereas the Virtual Factory is defined as a dynamic process-oriented model of a factory using models for processes, logistics and simulation. The authors term the cooperation of these static and dynamic models digital production, and they assign tools for e.g. CAD/CAM, CAPP to the Digital Factory and tools for process modelling or simulation and Virtual Reality tools to the Virtual Factory.

Virtual Reality (VR) is an essential concept in the context of the Digital Factory, and its most important feature is the close to reality 3D real-time visualisation. VR was envisioned already in 1965 by the computer graphics pioneer Ivan E. Sutherland who described it in "The Ultimate Display" (Sutherland, 1965), and three years later he presented a prototypical head-mounted display (HMD) system (Sutherland, 1968). Actually, the term VR describes virtual 3D worlds with different grade of immersion, allowing the user to intuitively interact (e.g. move or manipulate) with digital models in this virtual world as if they were real (Freund et al., 2001b, Wiendahl et al., 2003, Bal & Hashemipour, 2009). According to (Mujber et al., 2004) VR systems can be classified in three categories, depending on the grade of immersion:

1. Non-immersive VR with standard input devices (e.g. mouse, keyboard) and output on standard monitors
2. Semi-immersive VR with input devices such as space balls, joysticks and data gloves and output on large screen monitors or projection systems
3. Fully-immersive VR with user interaction via data gloves and voice commands and output on HMDs or projection rooms

Immersive VR can be regarded as a new type of man-machine interface by means of devices such as 3D stereo monitors, HMDs and data gloves. The use of large projection surfaces or rooms, up to now more commonly used than HMDs, is industrial practice for a long time (Gausemeier et al., 2000). The emerging new HMDs such as Oculus-Rift will probably be applied more often for industrial simulation and VC (McGregor, 2015, Digital-Engineering, 2016). A popular name for a projection room is CAVE (Cave Automatic Virtual Environment), defined by the NCSA (National Center for Supercomputing Applications) as *„an immersive virtual reality facility designed for the exploration of and interaction with spatially engaging environments“* (Wikipedia, 2014). This technology is supported by several 3D plant simulation systems such as CIROS (RIF, 2014), see also subsection 5.2.1. A comprehensive literature survey on 154 VR applications in development of manufacturing processes and systems is given in (Choi et al., 2015). The findings of Choi et al. (2015) and Mujber et al (2004) show that VR, primarily applied to product design only, can be used for many different application from planning and design, engineering up to operator training. Particularly with regard to simulation and VC, VR is useful because it supports the validation of simulation model behaviour, and on the other hand, it supports process verification (e.g. assembly process) by 3D visualisation and immersion if reasonable. The potential of VR in the context of controller programming validation has been already outlined in (Spath & Osmers, 1996). At that time, the authors proposed an interactive generation of PLC programs using a 3D VR environment, not VC.

The term “Digital Factory” is used since the mid 1990s (Bylinsky, 1994), but research started intensively after the year 2000. A literature review by (Himmler & Amberg, 2013) identified 45 definitions, from which the authors built three categories:

1. The Digital Factory as digital copy of an already existing real factory to represent the present structures in the IT systems is used by 10 definitions.

2. A broader conceptualisation is given by 19 definitions, where the Digital Factory is seen as a concept for the planning and validation of production and plant before realisation by the use of digital models and methods.
3. The third category with 16 definitions extends the second definition and additionally includes the preceding product design.

In the past few years the most common definition, fitting in the third category, is given by the VDI with the guideline 4499 (Himmler & Amberg, 2013). This guideline defines the Digital factory as *"the generic term for a comprehensive network of digital models, methods and tools – including simulation and 3D visualisation – integrated by a continuous data management system"* (VDI, 2008). This implies integrated planning, simulation and validation of manufacturing processes and systems by the means of digital models, computer-aided planning and design and computer-aided engineering with associated software tools and with the aid of an integrated data management.

The application of the "Digital Factory" is postulated to extend to all phases of manufacturing system planning and development (Fig. 2-7) by integration of software tools for planning, design, engineering and simulation (Kuehn, 2006b). Nowadays simulation is already used intensively in some phases (Kuehn, 2006a), e.g. part manufacturing (NC programming for machine tools), material flow simulation for layout planning with throughput optimization and facility design, 3D simulation of kinematics for mechanical engineering and possibly interaction with product data at Digital Mock-Up (DMU). Off-line programming and 3D robot simulation have been used for many years; sometimes even 3D human models are used today to evaluate ergonomics of manual assembly tasks in design (Caputo et al., 2006, Lämkull et al., 2009, VDI, 2015b). Besides ergonomic evaluation, an integration of virtual humans in the simulation of the material workflow during VC becomes possible by their programmability and ability to manipulate material (e.g. take workpieces from a transportation system) similar to an industrial robot (Rossmann et al., 2006, Schlette & Rossmann, 2009).

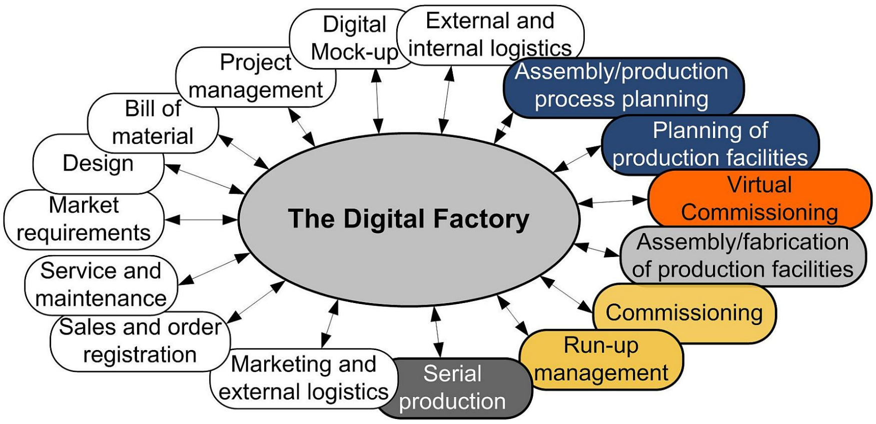

The Digital Factory
Digital Mock-up
External and internal logistics
Assembly/production process planning
Planning of production facilities
Virtual Commissioning
Assembly/fabrication of production facilities
Commissioning
Run-up management
Serial production
Marketing and external logistics
Sales and order registration
Service and maintenance
Market requirements
Design
Bill of material
Project management

Fig. 2-7: Digital Factory from planning up to the serial production derived from (VDI, 2011) with additional VC

Himmler & Amberg (2013) identified three main goals and their beneficial effects promised by the implementation of the Digital Factory:

1. Process lead-time reduction by e.g. increased planning speed and reduced ramp-up time
2. Cost reduction by e.g. decreased costs for planning, production and changes, easier and validated changes of production systems
3. Quality improvement by e.g. better planning quality and better reliability through validation

The main drawbacks and problematic issues stated in the publications analysed by Himmler & Amberg (2013) focus on lacking data integration, missing interfaces for data exchange in existing heterogeneous tool landscapes (no available standardised exchange format) and high costs for hardware and software.

Since the start of using simulation in the Digital Factory the focus has been primarily in planning, design and mechanical engineering (Schlögl, 2007). Production output, fit and specified behaviour of mechanical plant components must be ensured, but driven by automotive industry VC is recently getting more and more important, thus VC became an element of the VDI guideline 4499 Part 2. In (VDI, 2011) VC is localised as *"final step in the planning of the production equipment"* conducted *„before the real commissioning"*. According to this guideline (Fig. 2-8) VC is placed between the end of engineering and the beginning of manufacturing / assembly of the plant (cf. Fig 1-8).

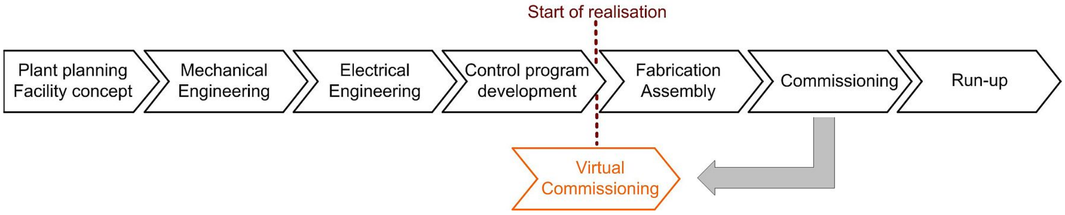


Fig. 2-8: Basic idea of VC according to VDI guideline 4499 Part 2 (VDI, 2011)

If all user-relevant data including a system model containing data of mechanics, electrics and information technology are available, the VC should allow the validation of e.g. *"planning results concerning cycle times or system availability"*. The guideline defines the configuration of a VC as a combination *of „one or more (depending on the commissioning progress) really existing electronic, hydraulic, pneumatic or other components, such as a controller or an operating panel, in a closed control loop with the virtual commissioning computer"*. The VC shall not only permit the test of PLC, NC or robot programs from OLP but also the *„evaluation of the machine behaviour in the future real system"*.

As advantage of VC, the guideline emphasises the early discovering of design, planning and engineering errors such as:

- Errors in the control code, e. g. logical errors, typos and copy-paste-errors
- Sequencing errors in the control code, leading to collisions (e.g. robot gripper and transportation system)
- Mistaken enable and interlock signals
- Misplanned system parameters, such as cycle time, quantity produced or availability

Additionally the VC shall validate the dynamic behaviour of the manufacturing systems, different operation modes (e.g. hand, automatic), exceptions and the targeted simulation of malfunctions such as broken cable or defective sensors.

Essential for VC is the creation of a virtual plant model reflecting the physical system with simulated mechanics (including actuator and sensor interfaces), connected to controllers (PLC, robot control) running the real software code (VDI, 2011). This is possible in a SIL or HIL configuration. This virtual plant model requires the incorporation of different CAE data from its components e.g. CAD drawings, wiring diagrams, I/O configurations and PLC/robot programs.

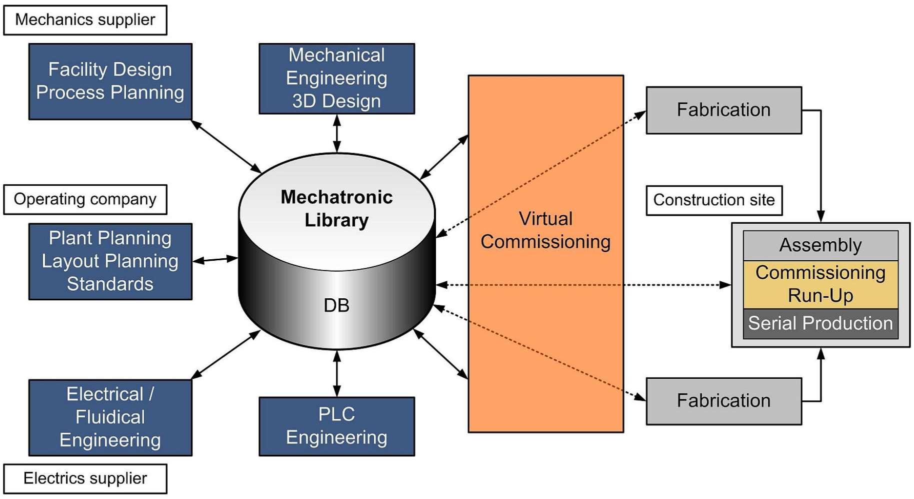


Fig. 2-9: Interdisciplinary cooperation based on a mechatronic library corresponding with (VDI, 2011)

The guideline points out the importance of an interdisciplinary cooperation between the different planning and engineering domains and postulates *"a holistic, mechatronic view of data on products, processes and resources along the production system's entire life cycle"*. For that purpose the guideline proposes to store all relevant data and models in a mechatronic library (Fig. 2-9) with interfaces to all planning and engineering systems including simulation systems for VC.

This approach requires an integrated data management with appropriate data formats and interfaces. Either with central database and central data model, for example hosted on a server in a corporate network, decentral data storage for the different crafts using a common data model or decentralised data storage and different data models (Schleipen et al., 2010).

Mainly in automotive industry the use of mechatronic model libraries (Fig 2-9) is currently emerging, but these libraries are designed for company internal use only and are build following user specific standards (e.g. Daimler Integra). They are not intended and not suitable for broad exchange of e.g. mechatronic component models (given as oral information by J. Kiefer from Daimler on a 2010 VDI/VDE GMA-FA 6.11 workshop at the University Dortmund), at most available for sub-contractors who are forced to do simulation based on the OEM’s standards and with prescribed tools.

This has not to be mistaken with a provision of tool independent mechatronic component models suitable for arbitrary simulation tools by manufacturers of standard components and subsystems such as e.g. FESTO proposed in this thesis (cf. subsection 5.4.2). Thus, these libraries are isolated applications in this respect today. Nevertheless, component suppliers or sub-contractors are forced to deliver e.g. CAD data or PLC programs following specific standards of large OEMs from automotive industry.

2.7.3 Application of simulation and VC in large companies

In the context of the “Digital Factory”, it is in principle possible to use the complex off-the-shelf factory planning and engineering suites of tools from the market-leading vendors (Dassault Delmia, Siemens PLM) and to conduct a VC based on a company specific mechatronic library. Generally, this requires the commitment for one of these suites of tools and then a change or the application of other commonly used tools is complicated. These suites of tools aim for the support of all phases of factory planning, but they usually require high investment costs, high deployment penetration (whole departments), a high-level of training or in-house secondment of consultants from the vendors for the implementation of custom-built functions and the laborious building of model libraries designed to the user’s specifications.

Therefore, mostly large companies (e.g. from automotive industry) selectively choose to conduct a VC using such suites of tools. Nevertheless the simulation models of complex manufacturing systems for VC are often not available in sufficient time to be justified for this purpose (Stern et al., 2010, Rossmann et al., 2012).

2.7.4 Application of simulation and VC in SME

Small and medium-sized enterprises (SMEs) are *"the backbone of the European economy"* (EC, 2015a). The EU defines the main factors for being an SME with a staff headcount < 250 employees and a turnover ≤ € 50 millions. According to this definition 90% of enterprises in the EU are SMEs which generate 67% of total employment (EC, 2015b). According to (Byrne et al., 2015) referencing 2012 data of the EU, more than 5.1 million companies are in business in the manufacturing and construction sector, most of all SMEs, to wit 99.6%. This great number of companies punctuates the relevance of problems elaborated below and shows the need for solutions for SMEs.

Most of the SMEs manufacture products in small lot sizes, thus already the profitable application of complex manufacturing systems with e.g. industrial robots poses a challenge. Especially because the commissioning of such highly automated systems tends to result in *"an unbalanced cost benefit ratio"* compared to less complex systems such as manual manufacturing. On the other hand, the introduction of VC is hampered by the necessary effort for simulation model building and additional costs for licences and maintenance, and thus the *"full potential of virtual commissioning is hardly reclaimable"* for SMEs (Brecher et al., 2013a). Mainly the situation of SMEs can be considered critical nowadays, because they also need cost-efficient and riskless methods and tools for modelling and simulation of complex manufacturing systems to implement necessary changes of existing systems without interrupting the running production (Bal & Hashemipour, 2009).

The availability and the advantages of VC are generally not well known in SMEs neither are the tools for VC, apart from perhaps, as part of the "Digital Factory" solutions for large companies or in the case of HIL simulation the method is supposed to be only for the development of ECUs. Consequently, there is only limited use of these solutions by SMEs; on the one hand because of the assumption to be applicable to ECUs only, on the other hand because they generally do not have the resources to start solving their pressing problems with such "Digital Factory" suites of tools (Westkämper et al., 2003, Drath et al., 2008a). Gu et al. (2007) and Schetinin et al. (2013) clarify why companies retain to use HIL; these reasons are particularly valid for SMEs and VC (cf. subsection 2.4.5).

HIL and VC have limited use in SMEs and simulation more generally is likewise relatively limited in its use. Byrne et al. (2015) identify the following reasons in the context of DES: complex process of simulation in relation to the small scale of the company, lack of data for simulation modelling, lack of time and resources required for modelling, lack of awareness and the underestimation of possible advantages by the use of simulation.

Bös (2008) references a study of the Fraunhofer IPA institute (Bierschenk et al., 2005) which states that 80% of large companies make use of simulation systems but less than 5% of small companies (<50 employees) use simulation tools.

In (Köbler & Pleuler, 2011) the authors report on a survey wherein SMEs were interviewed regarding their use of digital factory planning. The outcome show that 62,5% of participants do not employ an expert for factory planning, and 75% do not even make use of external know-how. None of these companies makes use of special tools from the field of "Digital Factory", but 90% of participants use CAD tool (even if mostly for layout design). The entry costs for the application (licences, training etc.) are very high – too high in the opinion of 58% of survey participants.

Besides the costs rated too high, these Digital Factory suites of tools are often too complex for SMEs to reasonably assimilate (Köbler & Pleuler, 2011, Bös, 2008), and the change needed, from their previous practice using simpler and independent tools to these suites of tools would not generally be plausible. A lack of 'easy to use' engineering and simulation environments impedes engineers in SMEs to set up and conduct VC.

Bös (2008) postulates the use of VR tools for 3D virtual plants with availability of comprehensive libraries. These libraries should provide adequate import possibilities that are necessary for the integration in heterogeneous system landscapes of SME.

The time pressure during projects prohibits the simultaneous introduction of new methods and tools and would, in conjunction with the lack of skilled experts, inhibit uptake and the prerequisite building of simulation model libraries. If the number of newly built production lines is too small, there would be no return on investment in the training of personal and the remarkable modelling efforts for one project. In addition, if a next project will be dissimilar to the first one, the modelling would often require again high efforts, as it will not allow the reuse of the already built library models. As a result of these facts, the level of use of VC in SMEs is rare (Stern et al., 2010).

Nevertheless, also SMEs are forced to do the planning and engineering with higher efficiency and better quality by improvements such as shorter project through-put time (faster "time-to-market"), reduced planning and engineering time, better planning quality and reduced costs for changing facilities, start-up and production (Bös, 2008). Hence, SMEs will be faced with the need for the partial introduction of methods from the "Digital Factory" at least, for example by introducing VC. Müür and Pettai (2010) propose efforts from universities e.g. by promotion of discussion groups or providing seminars to support the transfer of technology to SMEs for improved automation engineering.

They stress the importance of modelling and simulation of industrial automated systems and propose the implementation of tools like CIROS or ABB RobotStudio.

Essentially for this proceeding will be the implementation of 3D CAD (Bracht & Reichert, 2010). The use of 3D CAD is also necessary for simulation model building. Brecher et al. (2013a) also identified this as problematic, and named the *"generation of the virtual CAD work cell model"* as *"one of the greatest challenges concerning simulation-based robot commissioning today"*. If CAD tools are mostly used for 2D layout design so far, as it is often the case for SMEs (Köbler & Pleuler, 2011), this will be a problem, because 3D CAD is necessary to build the virtual manufacturing system model. To date, only 2D design drawings are often used, instead of 3D modelling because 'That's the way we've always done it' (Schulze, 2016). Hence, SMEs will be confronted by the need for the introduction of 3D plant simulation, and essentially will be the implementation of 3D CAD, if not already carried out. Thus, a substantial part of the research in this thesis has been devoted to a systematisation of work-flow from 3D CAD to mechatronic component models to allow for simulation model building suitable for SMEs (cf. chapter 5).

2.8 3D plant simulation tools

Today, more than a dozen industrial 3D simulation tools are available and applicable to VC. A survey carried out at the Automation and Control Engineering Group (Bockstette, 2013) investigated several simulation tools regarding their suitability for a VC of the manufacturing system to be used as test-bed (cf. chapter 4). The list of tools that has been reviewed more detailed in alphabetical order:

- ABB Robot Studio
- CIROS Studio
- Demo3D/Emulate3D
- Kuka Sim
- Simulation Wildlife
- Tara VRbuilder / VRcontrol
- TrySim
- Virtual Universe
- Virtuos (Virtuos M / Virtuos V)
- Visual Components 3D Create with PLC Add-on
- Xcelgo Experior

The review has been conducted with short-term licences or demo versions provided by vendors. The defined criteria for review had been:

- Documentation (manuals German/English, online help), support (support web pages, web forum, wiki, FAQs…)
- Available CAD data import possibilities via native/standard exchange formats

- Simulation model libraries (content and expandability)
- Modelling features / Usability
- Internal virtual controllers including support of programming languages for PLCs and robots. Support of native languages (SIL) or only generic languages?
- Coupling to external real controllers (HIL) or emulators (SIL) including communication technology (OPC…)
- Which kind of sensors are provided (measurement principles) and are they close to real sensors or e.g. only simulation sensors (detecting e.g. "closest object")

Tools with focus on material flow simulation such as Siemens Plant Simulation or AutoMod have not been reviewed in detail, neither the tools from the Digital Factory suites of tools, specifically Delmia Automation and Siemens Process Simulate & Commissioning. Besides, simulation tools which are mostly used in coupled configurations for VC of manufacturing systems such as WinMOD <-> INVISION, WinMOD <-> RF::Suite, SIMIT <-> 4Deploy or SIMIT <-> Mechatronics Concept Designer (MCD) have not been reviewed due to the requirement specified for this research, that the VC of the test-bed system shall be achievable with one simulation tool.

Generally, the reviewed simulation tools have their strengths and weaknesses, and a decision can only be made considering the manufacturing system to be simulated (cf. subsection 5.2). ABB Robot Studio and Kuka Sim (which is based on 3D Create) support only the simulation of robots from respective vendor i.e. ABB robots or Kuka robots, thus they have not been short-listed in this case. TrySim cannot import CAD models and provides only 2D modelling and simplified 3D view in simulation. Simulation Wildlife supports only DirectX (*.x) import, import tests of assembled components result in only one component in the simulation that cannot be edited or provided with kinematics. The matching of the remaining tools to the defined criteria results in the substantiation of the prior selection of CIROS and rated this tool again as one of the best suitable for the VC in this thesis. Finally, one interesting unique feature, Virtual Universe allows a parallel simulation of fluidics and electrics.

2.9 Data exchange with AutomationML

An ongoing interesting progression is the development of AutomationML (AutomationML, 2014) initiated by the companies Daimler, ABB, KUKA, Rockwell Automation and Siemens together with the Universities of Karlsruhe and Magdeburg and some smaller engineering companies in 2006, and published publicly first at the Industrial Fair in Hannover 2008 . The industry consortium opened up in 2009 and founded a registered association, the AutomationML e.V., which has, besides several industrial and academic members today. Since June 2014 the first part (architecture and general requirements) of AutomationML is an international standard (IEC-62714-1, 2014).

The very ambitious vision of AutomationML is *"the complete description of plants in a neutral format, providing all needed data for visualization and simulation targeting virtual commissioning"* (Weidemann, 2008). AutomationML is a neutral, intermediate data format based on XML for automation data storage and exchange including component model data, not limited to geometry (and kinematics if applicable) as this is the case for pure CAD exchange with e.g. STEP. The intention of AutomationML is the reduction of engineering efforts and quality improvement by interconnecting heterogeneous tools (Garcia & Drath, 2007), which may become especially valuable when setting up VC with different tools and exchange of model data using AutomationML. Drath, et al., (2008a) stated, *"Especially the phase of the virtual commissioning can utilize AutomationML"*.

AutomationML combines these already existing standards respectively data formats CAEX, COLLADA and PLCopen XML.

2.9.1 CAEX

CAEX (IEC 62424) is used as top-level format for the description of the plant topology and hierarchical structure of objects used in the manufacturing system (including necessary properties and relations between objects).

The development of the CAEX format started in cooperation of the RWTH Aachen (Epple, 2003) and the ABB research centre (Fedai et al., 2003). Further development was carried out with additional companies from process industry inside the German standardization committee DKE (DKE, 2014) and now CAEX is included in the international standardisation IEC 62424 (IEC-62424, 2008) "Specification for representation of process control engineering requests in P&I diagrams and for data exchange between P&ID tools and PCE-CAE tools."

CAEX is an abstract, neutral XML based data format supporting the storage of hierarchical object information, e.g. hierarchical topologies (plant, cells, devices, components) by applying object-oriented concepts such as attributes, data encapsulation, classes and class libraries, instances and instance hierarchies, interfaces, relations and inheritance.

The standard has been originally used for the description of process plants and for data exchange between planning tools and process control engineering tools (Mayr & Drath, 2007), but in (Güttel & Fay, 2008) the authors presented a first approach for its use in manufacturing engineering. Important elements are the "InstanceHierarchy" (hierarchical description of plant), "InterfaceClassLib" (signals between plant components or controllers, material flow, interfaces such as OPC), "RoleClassLib" (define roles as symbolic placeholder for plant components or functions) and the "SystemUnit-

ClassLib" (Library with device catalogues). In (Schleipen et al., 2008), CAEX has been used to transfer the necessary data of a small mechatronic plant consisting of two conveyors, a turntable and a test station for the automatic configuration of a production monitoring and control system.

Further examples for the description of manufacturing processes with AutomationML are given in (Schleipen & Drath, 2009, Lüder et al., 2010c).

2.9.2 COLLADA

COLLADA is used in AutomationML for describing the 3D geometry and kinematics with mechanical interconnections and dependencies. The approach to exchange kinematics between different tools (e.g. 3D CAD and simulation tools for VC) could be an important advantage compared to the exchange formats like STEP used today, where only exchange of geometry is implemented in tools (cf. subsection 5.3.2), if this would be widely accepted by developers of 3D CAD tools and 3D simulation tools.

According to (Grimm, 2011) COLLADA incorporates data such as part geometry (exact BREP and meshed model), definition of materials (shaders, textures) and the definition of product structure (parts and assemblies, mapping of materials to parts and LOD). Besides, possibly kinematic information such as the definition of kinematic models (description of joints, links and constraints), articulated systems (kinematic constraints, dynamic constraints), the assignment of kinematics to geometry and allows the assembly and parameterisation of kinematic systems.

2.9.3 PLCopen XML

PLCopen XML is incorporated for the description of overall behaviour (including electrical and control information like I/O relations). Regarding the term "behaviour", it has to be distinguished between the representation of internal behaviour of physical objects in e.g. component models and the description of e.g. PLC code for controlling physical objects (sequencing/logic), both is possible.

PLCopen XML provides an open interface for data exchange with other software tools, and has e.g. the potential to allow on the one hand the exchange of PLC projects between more different supporting IDEs in the future, and on the other hand, the standardised definition and exchange of logics/behaviour models for mechatronic components as part of AutomationML.

Since 2011, CODESYS V3 partly supports PLCopen XML as well as logi.CAD by logi.cals and MULTIPROG by KW-Software (PLCopen, 2011).

The working group for “Logic” of AutomationML e.V. has developed a plug-in based framework, the so-called logic CPF (Conditioner Pipeline Framework) programmed in C# (MS .Net). This framework allows the transformation of behaviour description given in the form of e.g. Gantt Charts, PERT Charts, State Charts and Impulse Diagrams to PLCopen XML (Estévez et al., 2010, Lüder et al., 2010a).

2.10 Conclusion

The literature review has presented several approaches to integrate different engineering domains like mechanical engineering, electrical engineering and controller programming.

For the specific research area of machine tools some approaches to integrate simulation in the engineering process are described, but there is no general solution for the design, off-line programming and simulation of manufacturing systems built of modular stations, sub-systems and components like sensor, actuators etc. supplied by different vendors.

The “Digital Factory” concept demands the use of complex planning and engineering suites of tools from the major vendors leading to very high training effort or in-house secondment of specialists from the vendor. Only large companies from e.g. automotive industry have the necessary resources available. This does not apply for SME, and there is therefore limited use of the Digital Factory.

The literature review has shown the potential of running a VC. An important goal of VC is the early offline verification and validation of control programs in conjunction with associated mechanical and electrical design, in order to reduce the considerable time delays during commissioning (Zäh & Wünsch 2005). These delays are often caused by the error-prone control program design. The benefit of Virtual Commissioning can be estimated on the basis of time, costs and software quality, at which the saying ”time is money” (Franklin, 1748) is valid as well, because reduced commissioning time will have positive effects on the costs.

The beneficial effects and advantages of VC such as reduced real commissioning time, higher quality planning and better control software quality are meanwhile reported by many researchers, e.g. (Auinger et al. 1999, McGregor 2002, Zäh et al. 2006, Reinhart and Wünsch 2007, Wischnewski 2007, Rossmann et al. 2007, Drath et al. 2008b, Starner and Chessin, 2010, Seidel et al., 2012), which is supported by reviews in e.g. (Hoffmann et al., 2010, Jain et al., 2010, Lee & Park, 2014).

This accentuates the need for VC, but often the modelling effort for the virtual manufacturing system is not been taken into consideration (if not neglected). It is judged by many authors to be difficult and associated with large effort (Moore et al., 2003, Park et al., 2006, Zäh et al., 2006a, Kain et al., 2009, Neugebauer & Schob, 2011, Ko et al., 2012, Park et al., 2013, Stich & Reinhart, 2013, Kim et al., 2013, Oppelt & Urbas, 2014), regardless of which simulation tool is used.

The outcome of this literature review shows the need for simplified or rather accelerated simulation model building to minimize the effort and necessary expertise required to build a virtual manufacturing system for V&V of control code and planned physical setup in a VC. The general need for such a plant model is stressed in (Lobo et al., 2013) *"The use of the approaches software in-the-loop…and hardware in-the-loop…, related with Simulation and/or Formal Verification – independently of advantages and disadvantages of each approach – has a common sensitive point that is the need of plant modelling"*.

The simulation model building could be supported by a mechatronical development of real components as proposed by e.g. Haq et al. (2010) which would also promote a cross-domain information exchange between different engineering domains (mechanical, electrical, control) that is lacking in current sequential processes (Eckes & Wagner, 2006). Such real mechatronic components could also ease the simulation building, because a decomposition and suitable modularisation would have been already done. A decomposition of a real manufacturing system has been for example demonstrated in (Lee et al., 2007). A standardisation of components with widespread occurrence is proposed in (Zäh et al., 2006a) to improve the reusability.

Modularisation is an important approach to handle the complexity of industrial systems, which is reflected in complex simulation model building (Meinert et al., 1999, Machado et al., 2006). If one has to start with an entire manufacturing system it is recommended by several researchers such as (Balci, 1990, Shannon, 1998) to decompose this system. Balci states, *"The system complexity can be overcome by way of decomposing the system into subsystems and subsystems into other subsystems"*, and proposes to examine how the system components are organised to prepare the decomposition. This is not necessarily an easy task. Shannon refers to Pareto's law with the statement *"that in every group or collection of entities, there exists a vital few and a trivial many. In fact 80% of the behavior can be explained by the action of 20% of the components"*. Hence, the problem in simulation model building can be the identification of the relevant components to be included in the virtual manufacturing system during System Definition and Model Conceptualisation (cf. Fig. 3-1).

Thus, to take the greatest advantage of this modular approach, this must comprise especially the set up and utilization of model libraries with standardised mechatronic component models in an environment allowing engineering, modelling and simulation, which is accessible even by SMEs. The use of model libraries is recommended by several researchers such as (McLean & Leong, 2001, Rossmann et al., 2007, Bös, 2008, Haq et al., 2010, Starner & Chessin, 2010, Seidel et al., 2012).

The complexity of industrial systems is still increasing and the effort for their commissioning tends to get progressively worse. Thus, engineers, and especially engineers in SMEs, urgently need guidelines for planning, implementation and reasonable execution of VC. From reviewed literature, no evidence was found of such guidelines or methodologies for VC.

The lack of methodologies is also stressed in (Harrison & Proctor, 2015) *"Research challenges are more evident in methods than abilities"* because available tools are principally capable to build a component-based simulation, but *"The optimal method for creating a component simulation is still up for debate"*. This has been stated in the context of "hybrid process simulation" (cf. subsection 2.7) where component means sub-systems such as conveyors, robots etc.

3 Proposals for VC methodology and a new simulation model building approach

Manufacturing systems are becoming more complex (cf. chapter 1). Thus, the increasing complexity of systems to be simulated can only be addressed *"by following a structured approach to conducting the simulation study"* (Balci, 1990), a conclusion that is still true for VC nowadays.

3.1 New proposal for a systematic VC simulation study methodology

After reviewing eight simulation study methods Montevechi et al. (2015) collected and described 21 general activities named in one or more approaches. The authors identified three main phases called "conception", "implementation" and "analysis" as spanning characteristic and subsumed all activities and approaches in a matrix table (Montevechi et al., 2015).

This review of Montevechi et al. and all other discovered proposals for simulation study methodologies are focused on simulation in general, or like in the named literature (see also chapter 2), they are focused on DES. A comparable methodology for VC is currently missing. Thus, this thesis proposes a new systematic VC simulation study methodology, intended to be notably adjuvant for SMEs, as general guideline for planning, implementing and conducting a VC for manufacturing systems.

This simulation study methodology (Fig. 3-1), specifically adapted to VC, has been concluded from the findings and proposals in (Shannon, 1976, Shannon, 1998, Banks, 1999, VDI, 2010, Law, 2009, Montevechi et al., 2015). The fundamental decision to apply simulation (VC) or not can be based on the 'pros' and 'cons' given in subsection 2.3. Differing from the depicted flow chart diagram, the data collection and preparation may be conducted concurrently with simulation model building.

In figure 3-1 this methodology is intentionally presented in a simplified way, reduced to the essentials to keep track of the overall process. For example, only the most important feedback arrows are drawn. Furthermore, it has not been defined by whom the steps or specific tasks from inside the steps are carried out.

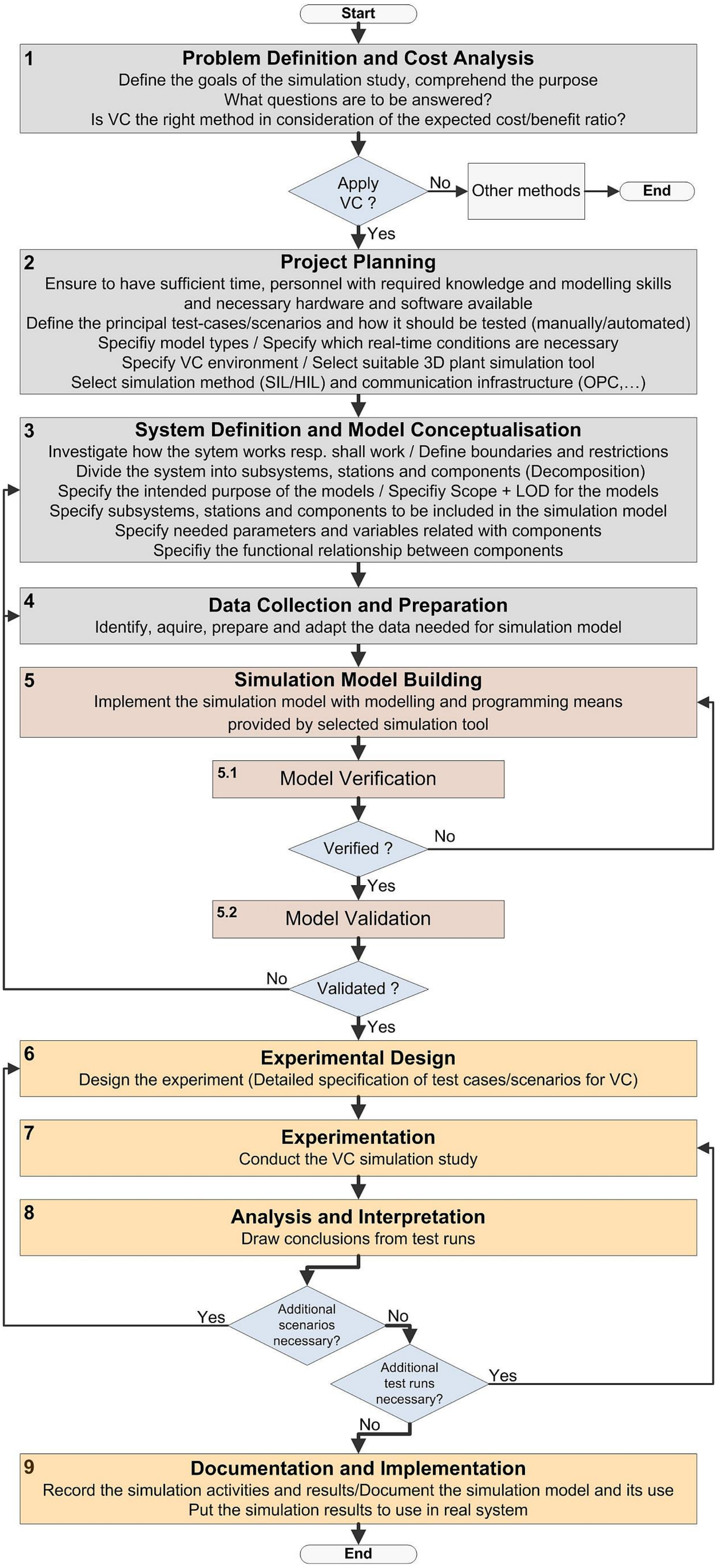

Start
1
Problem Definition and Cost Analysis
Define the goals of the simulation study, comprehend the purpose
What questions are to be answered?
Is VC the right method in consideration of the expected cost/benefit ratio?
Apply VC ?
No
Other methods
End
Yes
2
Project Planning
Ensure to have sufficient time, personnel with required knowledge and modelling skills and necessary hardware and software available
Define the principal test-cases/scenarios and how it should be tested (manually/automated)
Specifiy model types / Specify which real-time conditions are necessary
Specify VC environment / Select suitable 3D plant simulation tool
Select simulation method (SIL/HIL) and communication infrastructure (OPC,…)
3
System Definition and Model Conceptualisation
Investigate how the sytem works resp. shall work / Define boundaries and restrictions
Divide the system into subsystems, stations and components (Decomposition)
Specify the intended purpose of the models / Specifiy Scope + LOD for the models
Specify subsystems, stations and components to be included in the simulation model
Specify needed parameters and variables related with components
Specifiy the functional relationship between components
4
Data Collection and Preparation
Identify, aquire, prepare and adapt the data needed for simulation model
5
Simulation Model Building
Implement the simulation model with modelling and programming means provided by selected simulation tool
5.1
Model Verification
Verified ?
No
Yes
5.2
Model Validation
No
Validated ?
Yes
6
Experimental Design
Design the experiment (Detailed specification of test cases/scenarios for VC)
7
Experimentation
Conduct the VC simulation study
8
Analysis and Interpretation
Draw conclusions from test runs
Yes
Additional scenarios necessary?
No
Additional test runs necessary?
Yes
No
9
Documentation and Implementation
Record the simulation activities and results/Document the simulation model and its use
Put the simulation results to use in real system
End

Fig. 3-1: Proposed procedural steps for a VC simulation study

The attempt to include all possible feedback loops and all possibly involved company departments (e.g. mechanical/electrical engineering, PLC programming, HMI design, commissioning, IT, management etc.), staff members or external contributors respectively their roles would result in an overwhelming detailed diagram. The simulation model building, in whole or in part, for instance can be done e.g. by an existing staff member (who has to be trained then), by a newly hired simulation specialist or by an external service provider. Such a diagram could be designed for example according to the Business Process Model and Notation (BPMN) which provides e.g. activities, sequence flows, different gateways for decisions and merging, loops, data objects, groups and much more (OMG, 2016). Nevertheless, such a large diagram (which would span over several pages) could only illustrate one of many different possible ways to organise and divide the work. A company planning to start with VC could design a detailed procedure e.g. with BPMN according to company specific business processes, based on the proposed general methodology above and the more detailed modelling procedure proposals in subsection 5.3 where applicable.

The testing of this new proposal has been done by planning, implementation and execution of VC according to this general methodology by means of a test-bed. The detailing of the steps has been gradually defined and refined during the research carried out for this thesis.

The three main phases identified by Montevechi et al. (2015) can also be found in the methodology proposed here. The steps 1-4 belong to “conception”; steps 5, 5.1 and 5.2 form the “implementation” and steps 6-9 are among the “analysis”.

Shannon (1998) refers to the “40-20-40 rule“, which states that 40% of effort and time in a simulation project should be dedicated to the tasks from problem definition to data collection and preparation (Fig. 3-1, Steps 1-4). Only 20% should be spent for the implementation of the simulation model (Fig. 3-1, Step 5) and the remaining 40% for the steps from V&V up to documentation and implementation of results to the real system (Fig. 3-1, Steps 5.1-9).

Due to the mostly necessary considerable modelling effort, the goal to spent only 20% for simulation model building is often hardly to achieve today in the case of VC, particularly because commonly a multitude of components to be simulated are not yet part of the available simulation model library. The outcomes of the literature review in chapter 2 together with this rule lead to the following proposal.

3.2 Proposal for a new simulation model building approach

Concerning the conclusion of the literature review, the proposed new approach for simulation model building is aimed at the enlargement of simulation model libraries of arbitrary simulation tools as easy as possible, based upon modularisation respectively component-based mechatronical design.

In step 3 of the proposed general VC simulation study methodology (Fig. 3-1) the decomposition of manufacturing systems into sub-systems and stations is recommended according to the considerations of Reinhart and Wünsch (2007). This first stage of decomposition is done to subdivide the entire system into possible separate VC projects. From the starting point of this decomposition, a proceeding modularisation supports the new approach to generally split up the simulation model building (Fig. 3-1, Step 5) into the different modelling tasks “Low-level Component Modelling” and “High-level Plant Modelling” (cf. subsection 5.3.1).

The used test-bed, an existing manufacturing system at the UASA Hannover consisting of a transportation system and two robot cells, is presented in the next chapter. The first stage of decomposition of this entire manufacturing system into sub-systems will be presented and clarified as well as the further modularisation of the transportation system. This part of chapter 4 is related to step 3 (system definition) and lays the foundation for the simulation model building.

At the beginning of chapter 5, the requirements for VC will be discussed, which are linked to step 2 (project planning) and step 4 (data collection). Afterwards, according to step 2, the selection of a suitable 3D plant simulation tool will be established as well as the specification of the VC environment. Subsequently, the simulation model building, which is crucial for VC, will be analyzed. This is leading to a well-founded motivation for the general splitting into “Low-level Component Modelling” (cf. subsection 5.3.4) and “High-level Plant Modelling” (cf. subsection 5.3.5). Based on the test-bed, detailed proposals for both procedures are presented and validated in this thesis. This suggested principal splitting provides the basis to enlarge the existing libraries of 3D plant simulation tools with mechatronic component models provided by component manufacturers as recommended in this thesis (cf. subsection 5.4.2).

4 Test-bed for VC research

The increasing number of product types and product variants with simultaneous reduction of lot sizes and product life-cycle presented in the introduction (cf. chapter 1) involves the implementation of flexible manufacturing systems for the fabrication and assembly of such products. Such manufacturing systems provide adaptability to produce different products or product variants, either simultaneously or consecutively.

For a better structural overview, it makes sense to arrange such systems into groupings of sub-system:

- Manual or automatic part provision systems
- Flexible transportation systems with appendent carriers
- Robots, handling and machining systems with appendent tools and grippers
- Control systems (PLCs, CNCs and motion/robot controllers) and identification and data storage systems with mobile data carriers.

The operational flow of material can generally be managed by different means of conveyance such as manually by the worker, using a fork lifter or jack lift, conveyor belts, flexible transportation systems or automated guided vehicles (AGVs).

The interconnection of individual production and assembly line stations using an interlinking system for the automated transport of goods, including its associated control system, forms one variant of a manufacturing chain. It facilitates transportation between the stations it connects. Thus, a manufacturing chain can be an automated production line containing two or more production facilities with production flow controlled automatically by means of handling and control equipment, enabling continuous production according to the material flow principle. This is the usage subsequently adopted in this thesis. Depending on the type of interlinking system, it is possible to distinguish between rigid chaining and loose chaining (Warnecke & Schraft, 1984-1997).

Rigid chaining is described as the transport of goods from one work station to the next, carried out within fixed production cycles. The slowest workstation defines the unitary production cycle for the complete system. The disturbance of one workstation stops the complete line. Typical interlinkage systems for rigid chaining are clocked rotary transfer systems and clocked linear transfer systems.

In accordance with Warnecke and Schraft, one can describe loose chaining as the transport of goods from one workstation to the next, independent of the cycle times of the workstations. The individual stations work independently without a unitary cycle. The disturbance of one working station does not immediately affect the other workstations. A ‘stop’ on the complete system is delayed; buffer stores that can bridge short downtimes on single stations are possible. The layout and line-up of workstations is varied and flexible.

Flexible manufacturing systems are characterised by the use of loose chaining, mostly by means of longitudinal transfer systems, often generating the motion by friction between a carrier and a belt, plastic link chain etc. Different suppliers offer a broad variety of interlinkage systems for loose chaining, which can be categorised in three groups:

- Group 1: Longitudinal transfer systems with friction generated motion. These systems vary in technical specifications such as size, load capacity, type of drive, transport medium and bend behaviour (Wischnewski & Rossmann, 2010). Examples in the form of single-belt conveyors (e.g. Montech LTE), dual-belt conveyors (e.g. Montech LT40, Stein 300/700, Bosch Rexroth TS1/TS2plus/TS4), work piece carriers on load rollers (e.g. Stein 500), pallet transfer systems on friction rollers (e.g. MiniTec RMS, Krups LOGO!MAT, Bosch Rexroth TS5), flexibly segmented chain conveyors (e.g. MiniTec GKF, FlexLink X45/X65/X85, Bosch Rexroth VarioFlow) etc., represents only a small sample of choices of systems.

- Group 2: Track-bound self-driving work piece carriers (e.g. Krups LOGO!MAT E-CART on tracks with integrated power rails), monorail conveyors (e.g. Montratec Montrac, former Montech), intelligent (with microcontroller) battery-driven transport cars (e.g. Bosch CTS40/60, cf. subsection 4.3), electric overhead monorail conveyors, increasing in sophistication to track-bound autonomous robots (e.g. Servus GmbH, Autonomous Robotic Carrier ARC3) etc.

- Group 3: Automated Guided Vehicles (AGV). These AGVs in form of e.g. pallet trucks or fork-lifters moving automatically without human control along real or virtual guidelines. That means they find their way by means of mechanical, inductive, magnetic or optical markers or they follow programmed routes inside environmental software models using inertial (gyroscopic) navigation, dead reckoning, laser scanner or vision systems.

The latest developments in research are robots or handling systems on autonomous mobile platforms such as KUKA youBot (KUKA, 2014) or FESTO Robotino (FESTO, 2015b, FESTO, 2015c) which could lead to even more flexible interlinkage within manufacturing systems in future.

The UASA Hannover has such a flexible manufacturing system available. It consists of a SCARA robot cell, a gantry robot cell and an interlinkage system for loose chaining controlled by a PLC. This system serves as a technology demonstrator for production processes and has been used for students training in robot programming and control programming in recent years. The transportation system is implemented with battery-driven transport cars on passive tracks and belongs to group 2; and is a Computerised Transportation System (CTS), manufactured by Bosch. Nowadays longitudinal transfer systems with friction generated motion of carriers are widely used in industry (Wischnewski & Rossmann, 2010), but the other examples of group 2 show the industrial relevance of systems with track-bound self-driving work piece carriers.

The impetus for the research project presented in this thesis came from the unsatisfactory situation which exists for such “Robot based Flexible Assembly Systems” (RFAS), an example of which is the laboratory demonstrator established at the UASA Hannover. To date, the V&V of the collegiate programs in this system had not been possible. At least, not without running the complete system with its associated PLC and/or robot controllers and process hardware. This situation could result in a catastrophic outcome for the hardware as whenever an error occurs, plausible even at the hands of a professional programmer in industry, damage may result.

A search for ready to use solutions has shown that there are tools for CAD based off-line programming and simulation associated with state-of-the-art industrial robots but not for assembled hardware systems. This lack of ready to use solutions applies particularly to flexible manufacturing systems as a whole but also to separate transportation systems or separate robot cells (containing an off-the-shelf robot and purpose-built periphery) such as the SCARA robot cell of the RFAS. Engineers in industry are confronted with the same problems at commissioning when dealing with e.g. untested programs for different interacting controllers. However, compared with the lab at the UASA Hannover, the difficulties have a greater significance, not least because of the greater complexity of the industrial systems. The complexity of industrial systems is still increasing (cf. chapter 1), hence the problems in industry tend to get progressively worse, and engineers in SMEs particularly, urgently need guidelines for implementation and reasonable execution of VC. Such guidelines are currently missing, and one goal of this thesis will be the development and provision of indicative procedural methods, applicable to SMEs according to chapter 3.

4.1 Specification and choice of test-bed

In order to investigate Virtual Commissioning and to demonstrate how to avoid the needs and risks of real commissioning, a system of sufficient complexity and industrial relevance is required. This system should include industrially relevant components, sub-systems and software tools in order that general conclusions may be drawn. The test system at the RFAS (see below) is believed to be sufficiently complex and representative for such research.

Without prejudice to the fact that the development of the implemented transportation system (Bosch CTS40) dates from 1990s, it is well suited to illustrate the emerging problems that are encountered when realising VC. As indicated above, similar systems with "intelligent" carriers are available, which would result in modelling efforts for VC as challenging as the transportation system available at the RFAS. According to a personal communication from Bent Aksel Jørgensen, CEO from the Danish company Xcelgo, a large installation of a Bosch CTS60 system is still in use at LEGO Billund in Denmark. Xcelgo built a 3D simulation for this system using their tool Experior, which supports the use of such systems to initiate research on VC.

Accepting that the argument above supports the use of a manufacturing system such as the test-bed at the UASA Hannover can facilitate the VC research proposed for the project reported in this thesis, it was necessary to refurbish and enhance the functionality of the system as it was, in order to ensure the research conclusions could be more generally relevant. This preparatory work was subdivided into hardware and software reconstruction and enhancement tasks.

The manufacturing hardware system selected for the test-bed, i.e. the Robot based Flexible Assembly System (RFAS), consists of a transportation system (Bosch CTS40) with four active transport cars on passive tracks (approx. 20m) and two robot cells for assembly tasks, as shown in Fig. 4-1. The transport cars can enter both of the fully automated robot work cells. One robot cell is equipped with a Berger-Lahr gantry robot and the second robot cell, originally equipped with a Bosch SCARA robot (Fig. 4-1). The transportation system CTS40 of the RFAS is controlled by a PLC that also communicates with both robot controllers.

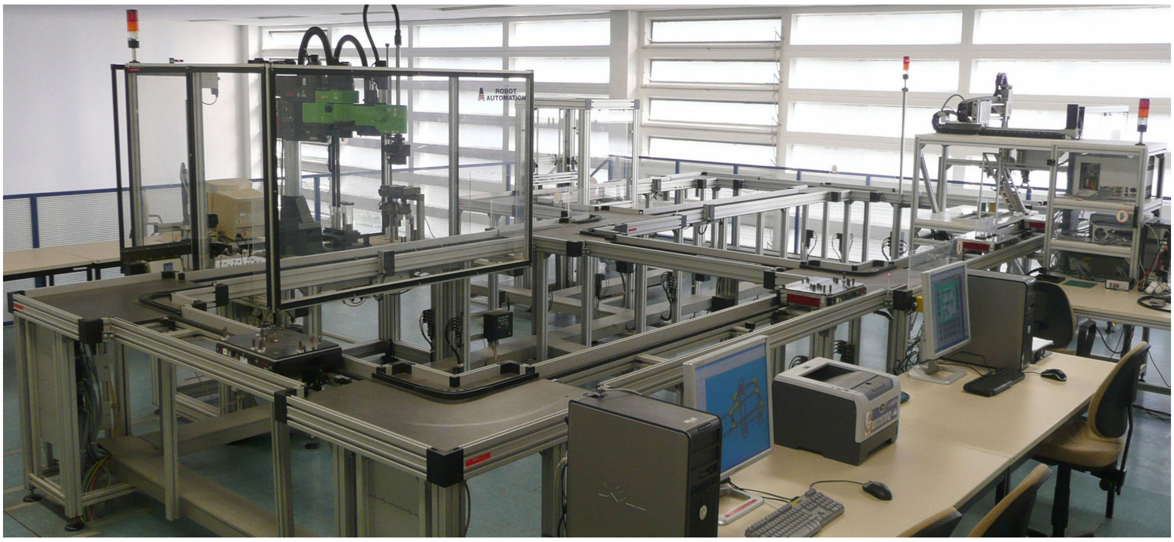


Fig. 4-1: Robot-based Flexible Assembly System (RFAS) at the UASA Hannover at project start

The system includes 5 curves, 4 track turnouts, 9 stopper units, 2 battery charging stations and a lifting-positioning unit inside the SCARA cell. Additionally it provides a system for transport car identification and data storage that consists of 6 data read/write stations on the tracks and mobile data carriers in the transport cars. The modelling and simulation of the entire manufacturing system is part of the ongoing VC research at the UASA Hannover, hence the complete RFAS will be described below to establish the context for elements considered in this study.

As mentioned, it made sense for a better structural overview to arrange the RFAS to sub-system groupings (Fig. 4-2). Besides associated PCs these sub-systems are:

- The central PLC with associated HMI panel
- The transportation system including appendant transport cars and identification/data storage system
- The gantry robot cell with robot, robot controller, automatic tooling system and parts provision system
- The SCARA robot cell with robot, robot controller, automatic tooling system and parts provision system

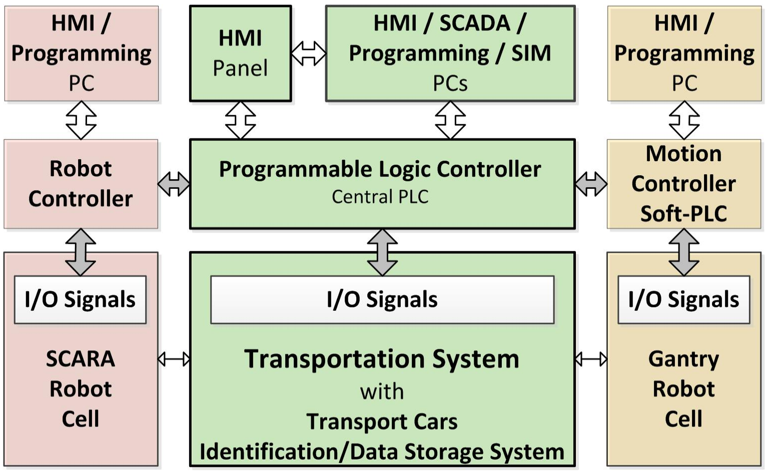


Fig. 4-2: Simplified structure of manufacturing system RFAS

These sub-systems belong either to the supervisory level and the control level, or to the field level respective process level, according to the well-documented classic automation pyramid (Fig. 4-3). Enterprise Resource Planning (ERP) software, Manufacturing Execution Systems (MES), Management Information Systems (MIS) or Executive Information Systems (EIS) are currently not in use at the RFAS.

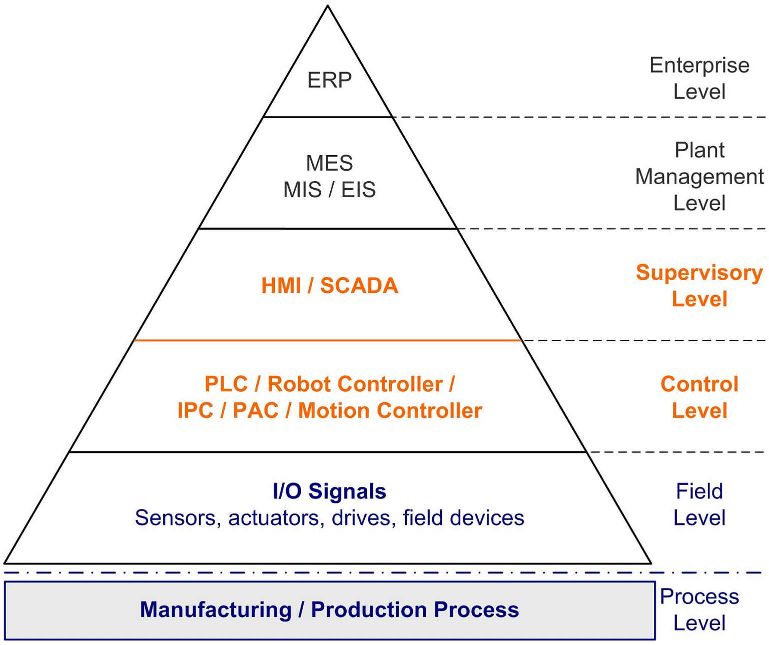


Fig. 4-3: Automation pyramid

The automation pyramid (Fig. 4-3) is based on DIN EN 62264-1 (DIN, 2014), but a variation from this standard of six hierarchical levels have been used here instead of five levels. The DIN standard considers the control level and supervisory level as one level (cf. Fig. 3, p. 17 in the standard). The objects to be analysed by VC are the HMI application on supervisory level and the PLC program on control level (orange), for this purpose objects at field level and process level (blue) have to be simulated.

Limiting the scope of VC in this thesis to the PLC with associated HMI system as objects for analyses; and the transportation system as the system to be simulated, is proposed as being sufficiently complex, with suitable physics' to test all stated hypotheses and research questions.

The VC research has been approached first through the modelling and simulation of the transportation system, as *"the integration of carrier-based transport systems is of special interest. This is because of their high complexity due to the huge number of applied sensors and actors"* (Rossmann et al., 2012). Moreover, the implemented routing strategies lead to complex program sequences requiring extensive testing of control programs (Wischnewski, 2007). An investigation of ramp-up processes by Denkena et al. (2008) indicates that flexible manufacturing systems with loose chaining tend to show more errors than systems with rigid chaining. Most of these errors resulting from control software and mechanical design have been detected in the first instance during real commissioning.

The VC of the SCARA robot cell and the VC of the gantry robot cell are possibly separate VC projects and are not in the focus of the project reported in this thesis. However, VC investigations for these robot cells are outlined in the recommendations for future work (cf. chapter 10) in which the findings of the VC of the transportation system following the proposed methodology will be taken into account.

This results in a gradual incremental approach toward the VC of the RFAS. The VC project reported in this thesis, comprising the transportation system and the PLC with associated HMI system, will be followed by separate VC projects for each robot cell in future work. This course of action correspondents to the decomposition recommended in procedural step 3 of the proposed methodology for VC simulation studies (Fig. 3-1).

4.2 Hardware reconstruction of the RFAS for VC research

The necessary hardware reconstruction of the manufacturing system at project start included the installation of a state-of-the-art PLC, which can be used conjointly and is compatible with advanced CAE and simulation tools for VC. At this time, a new controller for the gantry robot has also been installed. Integration into an environment suitable for conducting VC in the reported research project would not have been possible with the previous PLC and robot/motion controllers because of unavailable standard data interfaces and/or virtual controllers. Reconstruction of the SCARA robot cell is yet to be completed (cf. subsection 4.5).

4.2.1 Central Siemens S7-PLC

Figure 4-4 shows the newly installed Siemens S7-300 PLC system consisting of a Scalance Ethernet switch, power supply, CPU 317-2 PN/DP, two interface modules for double-row set-up, five digital input modules (32 channel) and four digital output modules (32 channel).

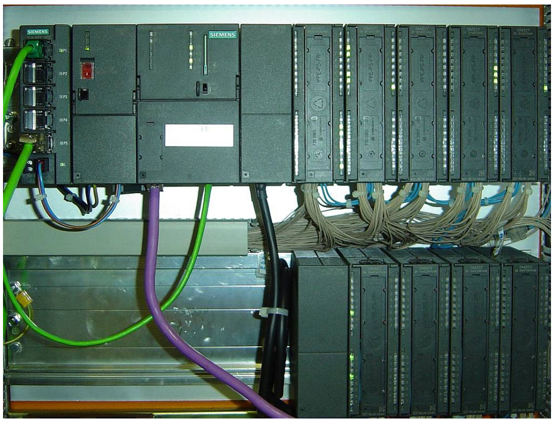


Fig. 4-4: Central PLC – SIEMENS S7-317-2 PN/DP

The selected CPU provides interfaces for Ethernet (Profinet-PN) and Profibus-DP besides the standard Siemens MPI bus. The PLC engineering is done with the associated software STEP7.

4.2.2 HMI touch panel

In order to improve the operator handling of the RFAS, a Siemens operator panel has been installed. This now predominantly used operator panel (OP 177 PN/DP) has a 4,3"color TFT touch display and additionally 32 programmable keys with LEDs (Fig. 4-5). The HMI engineering is effected using the associated Siemens software WinCC Flexible 2008 SP3 that allows the development of textual and graphical control displays for operation and visualisation of the RFAS. The operator panel is connected to the PLC and the engineering PC via an Ethernet switch (Siemens Scalance X005 - Fig. 4-6). The control displays developed are primarily used for the manual operation of the transportation system and for selection of different automatic programs.

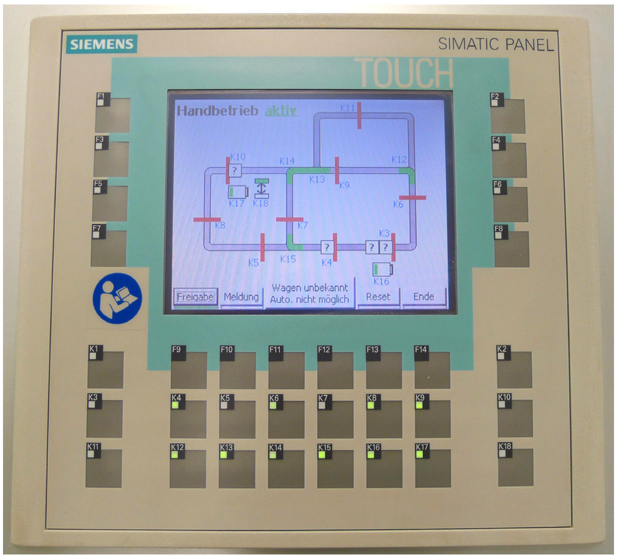


Fig. 4-5: Siemens operator panel with HMI for manual operation

In addition, it is possible to start a test program where one transport car inspects the complete track at which all sensors and actuators are tested and the result is displayed. An additional functionality is the selectable simulation mode for each of the robot cells that allows the simulation of robot I/Os by touch panel. Because the RFAS at present is predominantly operated by means of this panel, it makes sense to verify the HMI engineering and consequently it shall be part of the VC.

Figure 4-6 illustrates the current interlinkage and communication structure for the manufacturing system RFAS used as test-bed for the research on VC much more detailed than figure 4-2. Thus, the adequate complexity required to justify conclusions that are more general from the test-bed becomes clear and this illustrates the challenge of implementing and conducting a VC, even for such a comparatively small manufacturing system. The green boxes in the left hand part of figure 4-6 indicate the transportation system with associated PLC, operator/touch panel and required PCs for engineering, modelling and simulation, which are elements of the VC project presented in this thesis. The PLC, the operator/touch panel and PCs are connected via Ethernet. This limits the scope of this thesis to the PLC, HMI system and the transportation system with its microcontroller equipped transport cars, and is regarded as being sufficiently effective to develop and validate viable methodologies.

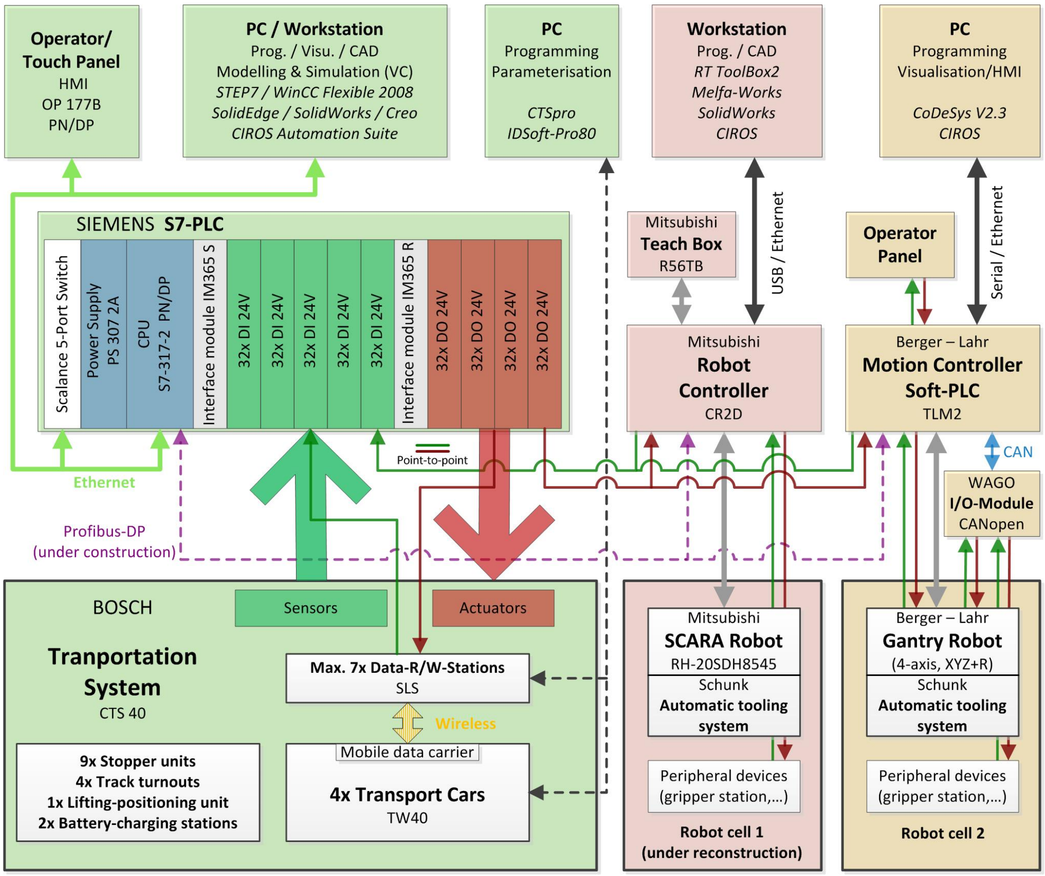


Fig. 4-6: Detailed structure of manufacturing system RFAS

Due to the exclusion of robot cells, the interaction of PLC and robot controllers has to be simulated by other means for the first stage of VC in this thesis. This has been solved by selectable simulation modes for the robot cells inside touch panel programming. The physical elements included are suitable for answering the current research questions, testing of the stated hypotheses and making the inclusion of robot cells unnecessary in reaching the generalised conclusion. Nevertheless, the developed environment is also suitable for the future work. The red boxes indicate the Mitsubishi SCARA robot with controller, teach box and associated workstation, whereas the orange boxes indicate the gantry robot cell with associated PC. These sections indicate the scope of two separate VC projects outlined in the recommendations for future work. The PLC and both robot controllers are connected point-to-point, but a Profibus-DP connection is under reconstruction.

Figure 4-7 shows the current layout of the transportation system with adumbrated robot cells.

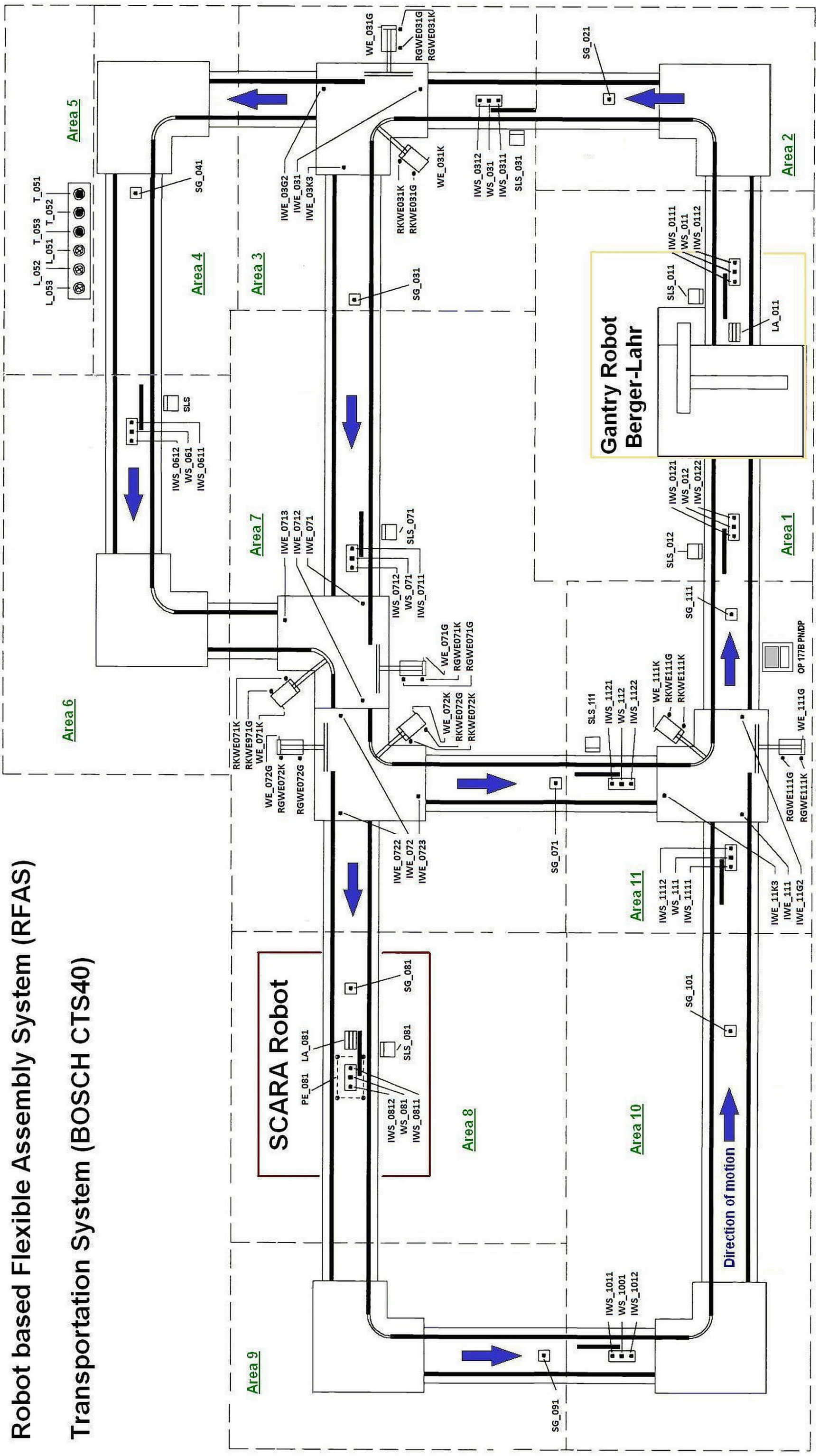


Fig. 4-7: Current layout of transportation system with adumbrated robot cells

The CAD model of the transportation system produced during the study is shown in Fig. 4-8.

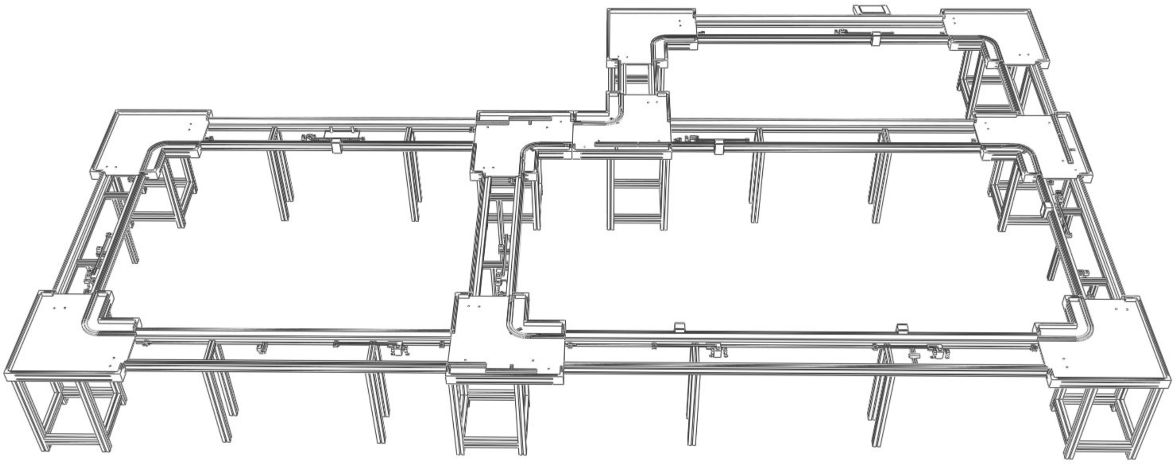


Fig. 4-8: 3D CAD model of entire transportation system

For an understanding of the transportation system relevant to subsequent modelling and simulation, it is important to establish the function of components and sub-system. Thus, the transportation system has been decomposed into components and sub-systems that are described below in detail. Furthermore, this detailed information will be adjuvant for the comprehension of the advisability regarding the provision of simulation models or at least 3D CAD models suitable for simulation by the manufacturers of components or sub-system, as proposed in a new approach (Subsection 5.4.1). The proposed advantageous provision makes all the more sense as increasing complexity of geometrical structure and/or logical behaviour of components or sub-systems becomes evident.

When starting implementation of VC, the modelling and simulation of an existing system should be helpful in gaining knowledge of the model building and simulation procedures and software tools and for the V&V of resulting simulation models. A good opportunity would be e.g. a planned redesign of the layout of a manufacturing system and its transportation system. It is recommended that such an opportunity be grasped, if VC has been considered to be worthwhile by company-internal decision makers (based on ‘pros’ and ‘cons’ as listed in subsection 2.3) during problem definition and cost analyses (cf. Fig. 3-1, step1): and to start its implementation based on a functioning system.

The relationship of the sub-systems HMI, PLC and the transportation system separated into tracks and transport cars, which are relevant to the VC study in this thesis, is illustrated in figure 4-9.

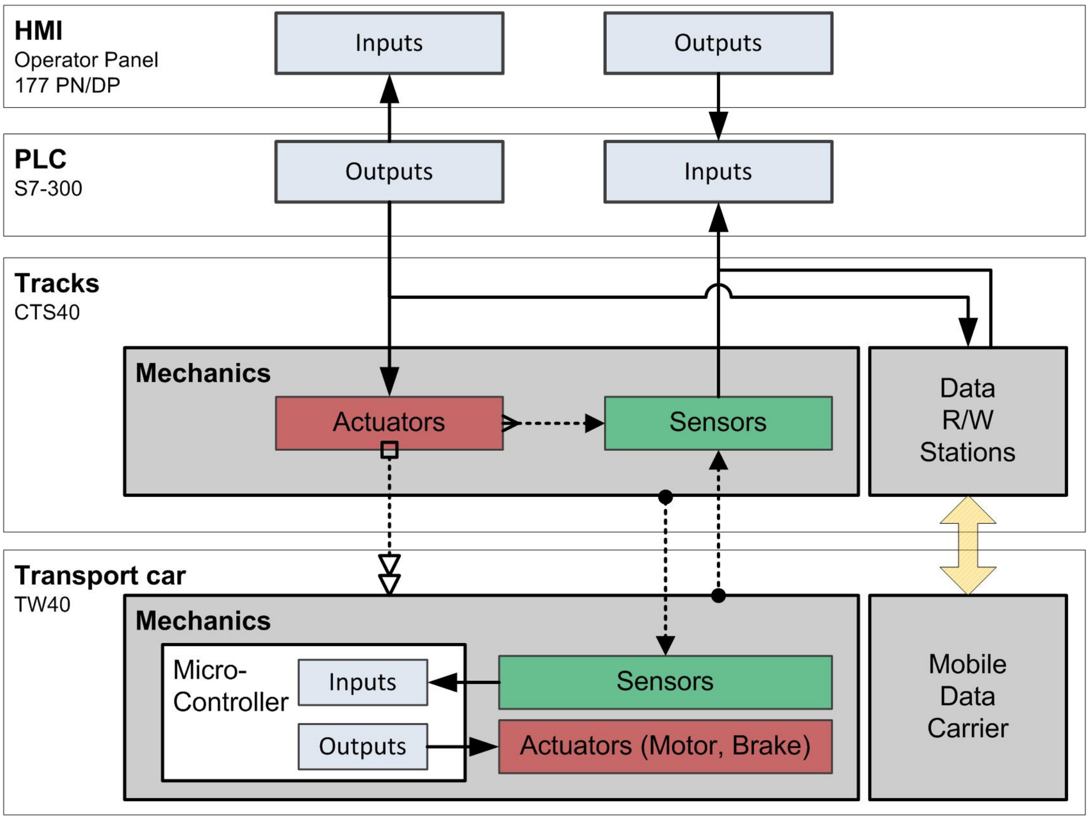


Fig. 4-9: Relationship of the sub-systems relevant to VC

A goal of the VC study in this thesis will be to investigate how the V&V of HMI control displays and PLC programs may be set up for the given transportation system in a systematic way. Therefore, all the different relationships between the sub-systems tracks and transport cars depicted in figure 4-9 have to be included in the simulation model of the transportation system (cf. Fig. 5-10). The VC has to incorporate either real or virtual HMI and PLC and a virtual transportation system containing mechatronic models of tracks and transport cars.

4.3 Transportation system – detailed description of test-bed

The transportation system Bosch CTS (Computerised Transfer System) was developed by Bosch as a modular system, customisable for the individual requirements of different customers. For this purpose a catalogue of basic coordinated components for two different sizes of transport cars were available: CTS40 for transport cars TW 40 with 20 kg payload and CTS60 for transport cars TW 60 with 40 kg payload. The combination of these components allows for the setup of manifold track layouts and functionalities. The catalogue includes passive mechanical components as well as active electrical components and electrical components with pneumatic auxiliary energy for movement of actuators.

The objective of the later modelling in this thesis is the creation of simulation models for the transportation system. Both robot cells will be modelled during the future work. The transportation system can first be decomposed into “Transport Car” and “Tracks” (Fig. 4-9). The “Tracks” subsumes the CTS40 components of the transportation system listed below:

<u>RFAS</u>

- Transportation system CTS40
    - Transport Car
        - Mechanics
        - Actuators
        - Sensor system
        - Control logics / Data storage
    - Tracks
        - Passive track components (only static mechanics)
            - Straight-line tracks with base frames
            - Curve components (90°) with base frames
            - Signal bars and code bars
        - Active track components (mechanical/electrical actuators / sensors)
            - Electro-pneumatic components
                - Track turnouts (left/right) with base frames
                - Stopper units
                - Lifting – positioning unit
            - Electric components
                - Data R/W stations
                - Inductive proximity track sensors
                - Battery charging stations
- SCARA robot cell
- Gantry robot cell

UASA Hannover uses the CTS 40 without any 180° curves, track turnouts in T-configuration, lifts or box changeover units that are additionally part of the Bosch CTS hardware building set.

A detailed description of the transport car TW 40 and the other individual components used for the tracks of the CTS40 system as part of the RFAS follows below. The detailed analysis of the system correspondents with the suggested investigation into how the system works in procedural step 3 of the proposed methodology for VC simulation studies (Fig. 3-1) and is important for an adequate modelling. The detailed description is useful to understand the modelling carried out to provide the basis for the subsequent simulation and VC of the transportation system. It is part of the modeller’s job *“to learn as much as possible about how the real system works”* (Carson II, 1989). It is essential to understand the system with its entities and technical constraints well enough to build an accurate simulation model (Sturrock, 2012).

The interaction between the transport car with its sensor system and sensors on the tracks, the wireless communication between data R/W stations on the tracks and mobile data carriers in the transport cars, along with the PLC, are especially important for a realistic simulation. The mechanics of the transport car activates inductive proximity sensors on the tracks as well as the mechanics of the tracks (signal bars/code bars) activates sensors of the transport car (Fig. 4-9).

The components and sub-systems presented below were created as parts of a CAD library during the study reported in this thesis. This library of CAD models (cf. chapter 6) provides the basis for the developed library of simulation objects.

The transport car is the most complex sub-system of the Bosch transportation system, not because of its geometry but because of several sensors and built-in user parameterisable microcontroller generating a versatile behaviour. The illustrations and textual information from technical manuals need to be interpreted in the modelling of the transport car and other components of the CTS. The information and interpretations given below, are based upon parts of the data available in manuals provided by the supplier (BOSCH, 1992a, BOSCH, 1992b).

Such an interpretation from a description given in vendor manuals is an error-prone procedure. The validation of a simulation model for a complex sub-system is necessary. Generally, it is most likely only possible to validate the model, based on interpretations and assumptions, when the equivalent hardware system is available. This hampers the realisation of a VC and does not meet the claim that a VC should be possible without having the hardware available.

These problems increase for more complex industrial systems, because for *“large-scale real systems, it is seldom the case that one individual understands how the system works in sufficient enough detail to build an accurate simulation model”* (Carson II, 1989).

4.3.1 Transport car TW 40

Four transport cars TW 40 are operating on the tracks of the RFAS. These electrically driven transport cars are used to supply both robot cells with component parts and to remove the finished assemblies. It is possible to adapt user specific mounting plates on top of the chassis of the transport cars, as exemplarily shown in Fig. 4-10.

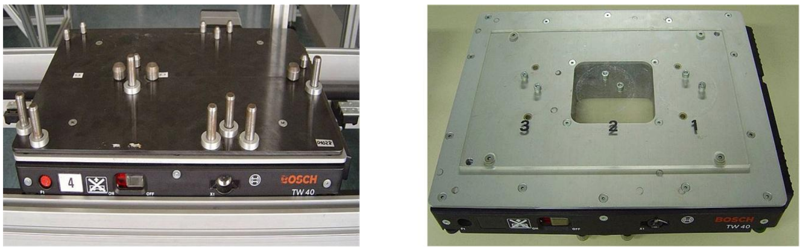


Fig. 4-10: Transport cars Bosch TW40 with different mounting plates

The TW 40 is powered by a brushless, electronically commutated (EC) motor with gear and electromagnetic brake. The motor drives both axles and thereby all four wheels via a toothed belt. It is equipped with accumulator, microcontroller, several sensors, seven-segment status display and a mobile data carrier. The control of drive and brake is not the sole task of the microcontroller. It also monitors the charge condition, manages the sleep modes and actuates status messages via the seven-segment display. The behaviour depends on an analysis of the built-in sensors and the set-up of speed, acceleration and other parameters. It is possible to read/write data from/to the mobile data carrier on the left hand side of the transport car (Fig. 4-11) by utilisation of data R/W stations (Fig. 4-21) alongside the tracks.

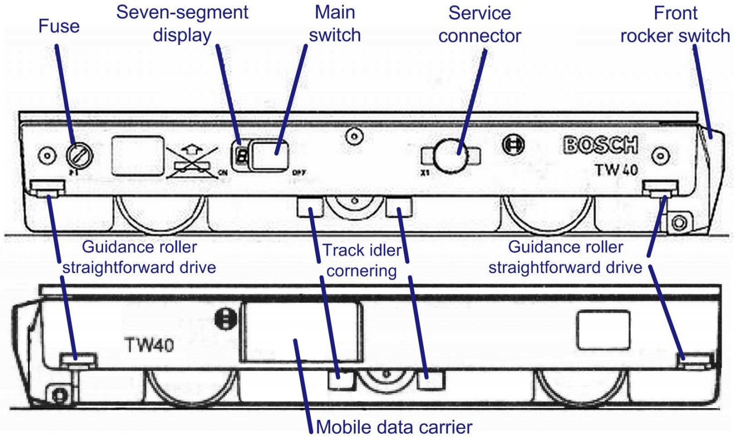


Fig. 4-11: Transport car TW 40 side views from (BOSCH, 1992a) – reproduced with courtesy of Bosch Rexroth AG

The service connector (Fig. 4-11) allows for the parameterisation of microcontroller software with different values for speed, acceleration and deceleration (see table 4-1) using the Bosch software tool CTSpro (BOSCH, 1992c). In addition, it is possible to charge the accumulator with an external power pack via this connector.

Tare weight:	11 kg		
Payload:	max. 20 kg		
Useable surface:	300 mm x 400 mm		
Motor:	Brushless EC motor		18 VDC / 100 W
Brake:	Electromagnetic brake		24 VDC / 6 W
Accumulator:	NiCd accumulator		24 V / 2 Ah, fast chargeable
Speed:	V_{min}	0,05 m/s	
	V_{max}	1,5 m/s	
	V_{low}	V_{min} …0,15 m/s	Speeds programmable via software CTSpro
	V_{sys} / V_{alt}	V_{low} …V_{max}	
	V_{cod}	V_{low} …1,0 m/s	Speed for cornering
Acceleration:	acc_{sys}	0,1 – 3 m/s^2	Acceleration depends on payload, programmable via software CTSpro
Deceleration:	dec_{sys} / dec_{alt}	0,1 – 3 m/s^2	Deceleration programmable via software CTSpro

Table 4-1: Technical data of transport car TW 40 (BOSCH, 1992a)

The seven-segment display (Fig. 4-11) next to the main switch (on-off) informs the user via several alphanumerical codes (some blinking at different frequency) about many different normal states or errors during operation (actual speed type, sensor events, charge and accumulator status etc.). Besides indicating the operational status of the TW, it is necessary during parameterisation with CTSpro.

The driving direction of the transport cars is given by the tracks. The transport cars can only move forward, but with different programmable velocities. They drive counter-clockwise on the tracks of the transportation system (Fig. 4-7). The transport car is not able to steer or to change independently its moving direction, thus suitable guiding track devices are necessary for low friction movement. For this purpose, the chassis of the transport car comes with guidance rollers for forward drive and track idlers for cornering. In straightforward drive conditions, these guidance rollers at all four corners prevent the transport car from wedging. The lateral track idlers are necessary for the cornering. The curves and track turnouts provide special guiding profiles at the inside lane, and the two lateral track idlers thread into these profiles and guide the transport car around the curve. In doing so, a ramp inside the guiding profile lifts the transport car at the inside lane, and only the small support wheel in between the track idlers and both drive wheels on the outside lane have contact with the track. This prevents slip and excessive frictional loss because the four-wheel drive; TW 40 has no differential gear. The four guidance rollers, as well as the track idlers with the support wheel lying in between, are also shown in the bottom view of the transport car (Fig. 4-12).

The following figure also shows both diagonally placed locating bushings for the location of the two tapered locating pins for the lifting-positioning unit (Fig. 4-26). The steel bars pictured, mounted on the solid aluminium chassis, are used to activate the inductive proximity sensors inside the track turn-outs (Fig. 4-22). The built-in accumulator is charged during operation by connecting the charging contacts shown in the underbody and the charging adapter.

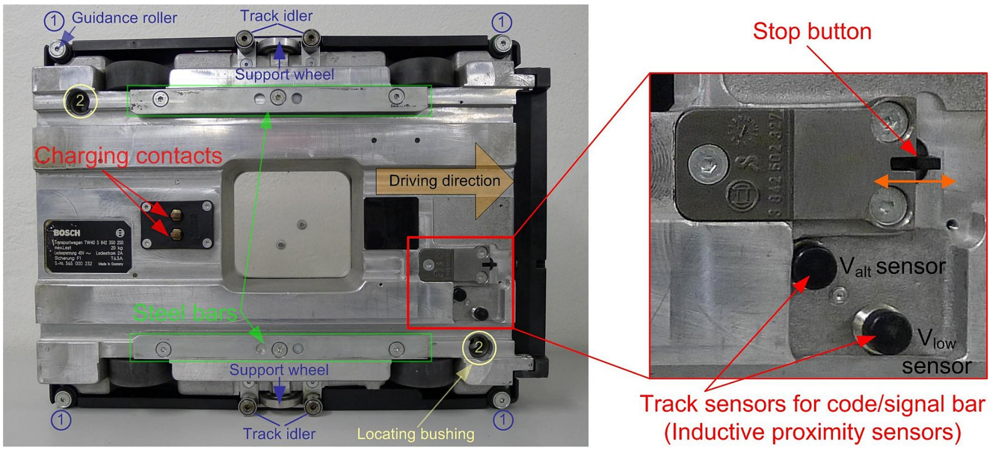


Fig. 4-12: Underbody of TW 40 with cutaway view of track sensors and stop button

The TW 40 is equipped with several sensors. The cutaway view shows the track sensors and the stop button in detail. The inductive proximity track sensors are used to switch different programmable speeds depending on the detection of the signal bar or the code bars (see Fig. 4-18 to 4-20) on the tracks. The signal of the sensor for minimum speed (V_{low}) invokes the deceleration parameter (dec_{sys}) and slows down the TW 40 to programmed minimum speed (0,05 m/s $\leq V_{low} \leq$ 0,15 m/s). This is only applied to stopper units at the RFAS. Behind the signal bar, the TW 40 would speed up with system acceleration (acc_{sys}) to system speed (V_{sys}), if it were not stopped by driving against the lifted stopper and thus pressing the stop button.

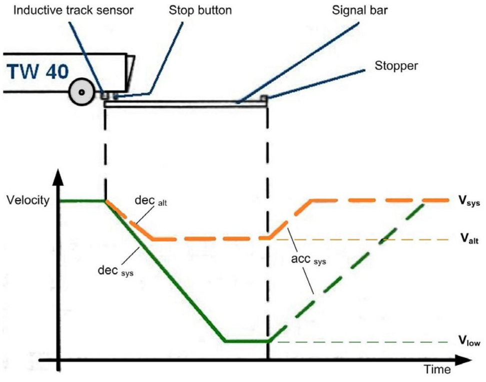


Fig. 4-13: Speeds of TW 40 depending on sensor signals (BOSCH, 1992a) - reproduced with courtesy of Bosch Rexroth AG

The signal of the sensor (Fig. 4-13, coloured original drawing) for alternative speed (V_{alt}) would invoke the alternative deceleration parameter (dec_{alt}) and would slow down the TW 40 to programmed alternative speed ($V_{low} \leq V_{alt} \leq 1{,}5$ m/s). This is not applied to the RFAS.

The code bar type A is a combination of two bars that activate both track sensors, and due to its special form it generates the pulse sequence shown in figure 4-14.

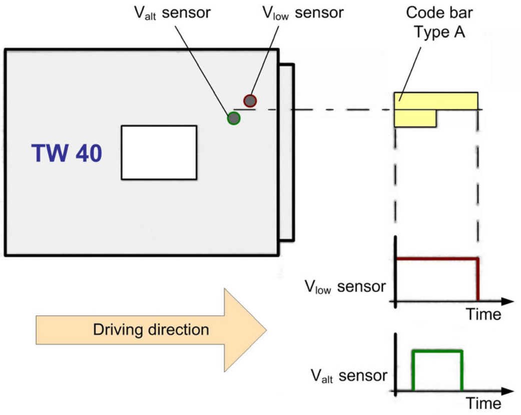


Fig. 4-14: Pulse sequence of track sensors when passing over code bar type A (BOSCH, 1992a) - reproduced with courtesy of Bosch Rexroth AG

This pulse sequence from both sensors invokes the coded speed ($V_{cod} \leq 1{,}0$ m/s), the programmed safe speed for cornering. This is applied before curves or turnouts at the RFAS. Additionally, the microcontroller ignores the optical distance sensors. These infrared distance sensors (Fig. 4-15) detect other transport cars running or standing ahead, and to a limited extent other obstacles too. In this case, the TW 40 will slow down to minimum speed (V_{low}).

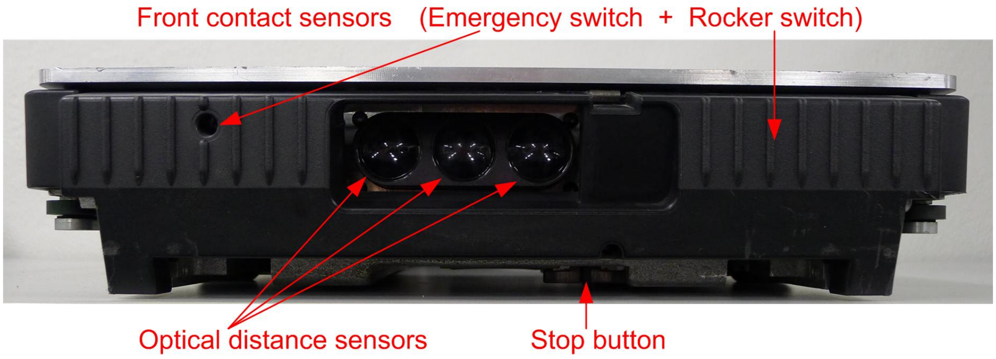


Fig. 4-15: Front view of TW 40 with sensors

A sensor detection distance of 500 mm or 1000 mm is selectable via CTSpro. If the obstacle disappears from viewing range of sensors the TW 40 will accelerate again, otherwise it will drive against this obstacle, usually another transport car, with minimum speed. This case activates the front rocker switch and the microcontroller switches off motor and brake, until the front rocker switch is released again. In the event of driving against an obstacle with higher speed (not specified), the front rocker activates an emergency switch leading to permanent drive switch-off and the notification of crash error “A” on the seven-segment display.

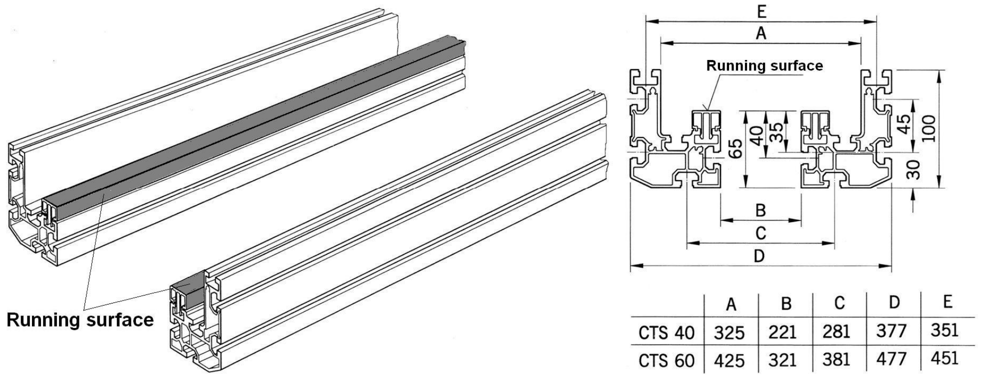
Additionally, the TW 40 provides a shock sensor, and negative acceleration > 2g similarly results in permanent drive switch-off and the notification of crash error “A”.

4.3.2 Tracks – Passive track components

Passive track components contain only static mechanics, and the RFAS includes straight-line tracks with base frames, curve components with base frames, signal bars and code bars.

4.3.2.1 Straight-line tracks

The basic construction of the CTS40 consists of tracks with ca. 250 mm track width. The special straight-line tracks (Fig. 4-16) and all base frames and pillars etc. are made of extruded aluminium profiles with slots for T-shaped sliding blocks for the assembly of other profiles or other attachments. Plastic profiles are snapped into the aluminium as running surface for the driving wheels of the transport cars.



	A	B	C	D	E
CTS 40	325	221	281	377	351
CTS 60	425	321	381	477	451

Fig. 4-16: Aluminium profiles with plastic running surfaces for CTS tracks (BOSCH, 1992b) - reproduced with courtesy of Bosch Rexroth AG

The tracks have to be braced by pillars with interspaces of less than 2 m. The aluminium border profiles of straight-line tracks and curves allow the use of slot-in acrylic or other sidewall material for personnel protection against accidental contact.

4.3.2.2 Curve component

The CTS curve component (Fig. 4-17) can be used as left turn or as right turn. It is a pure static component without actuators or sensors. The special guidance profiles, which are mounted on a synthetic resin base plate with plain surface, guide the transport car with reduced speed around the curve.

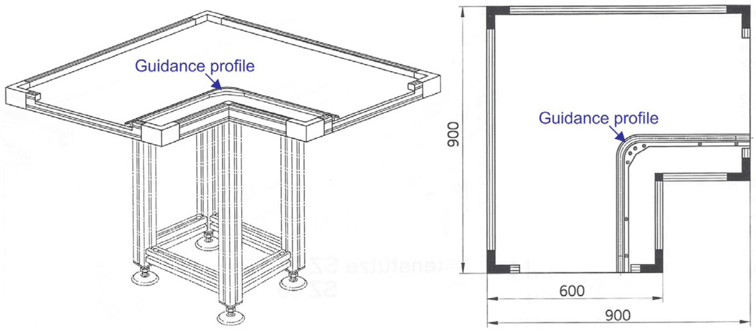


Fig. 4-17: CTS curve component (BOSCH, 1992b) - reproduced with courtesy of Bosch Rexroth AG

Inside the RFAS, five curves 90° are used as left turn to realise a counter-clockwise traffic.

4.3.2.3 Signal bar and code bars

Other passive components are the different code bars and signal bars. These components give notice to the transport car of the impending course of the track via two inductive proximity sensors in the underbody (Fig. 4-12). The signal bar (Fig. 4-18) brings the transport car to slow down when it is detected by the V_{low} sensor. The signal bar is mounted directly ahead or in combination with a stopper unit to reduce the speed of the transport car to programmed minimum speed (V_{low}) before it is stopped completely by pressing its stop button (Fig. 4-12). It is possible to use the signal bar at other track sections if lowest speed is necessary there.

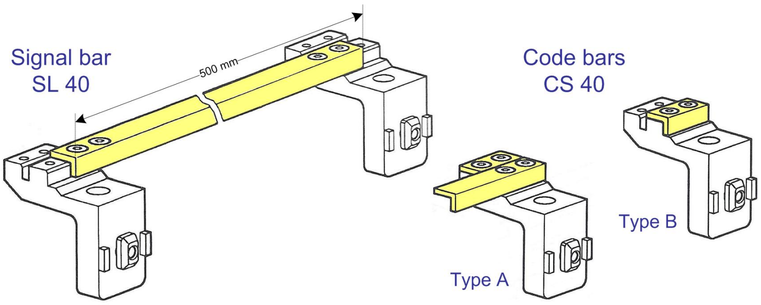


Fig. 4-18: CTS signal bar SL 40 and code bars CS 40 (BOSCH, 1992b)- reproduced with courtesy of Bosch Rexroth AG

The code bars are mounted as combination of a short and a bit longer bar (Fig. 4-18, Type A) at transition curve or at start of turnout to tell the transport car to slow down to safe curve speed (V_{cod}) and to ignore the optical distance sensor. The non-observance of optical distance sensor is necessary to prevent the transport car from slowing down to minimum speed (V_{low}) due to detection of framings, sidewalls etc. Because of the deactivated optical distance sensor the PLC program has to assure that only one transport car is located in the curve area at a time. A following transport car must not get the curve permission until the first transport car reaches the next inductive proximity track sensor.

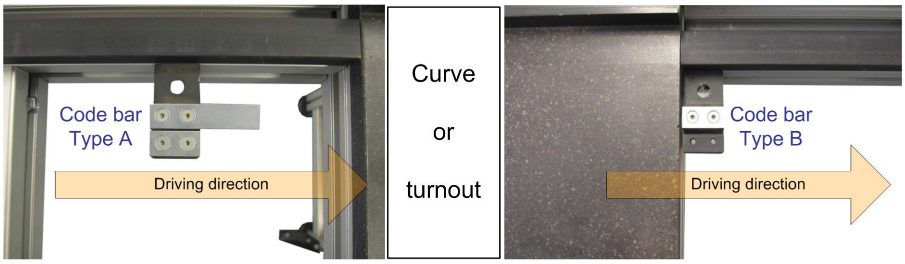


Fig. 4-19: Code bars used at RFAS

The single code bar (Fig. 4-19, Type B) behind the curve or turnout resets the reduced curve speed to the primary value and reactivates the optical distance sensor.

4.3.3 Tracks – Active track components

The RFAS contains the electro-pneumatic components (with mechanical movement) stopper unit, track turnout and lifting-positioning unit and the electric components data R/W station, battery charging station and single inductive proximity sensors.

4.3.3.1 Stopper unit (separator)

The stopper unit consists of a signal bar and inductive proximity sensors prior to the actual stopper (Fig. 4-20) and behind it. The stopper is a small air actuated lift cylinder controlled via 1 bit by the PLC. Setting the bit drags down the stopper, resetting the bit or a loss in air pressure lifts the stopper. Nine stopper units are mounted at different track positions. They are used inside both robot cells, in transport car holding areas 1, 6 and 10 (Fig. 4-7) and in front of all turnouts.

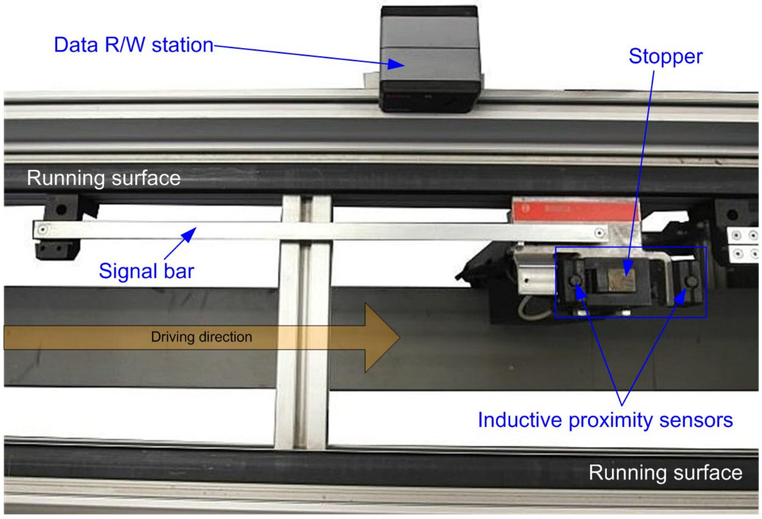


Fig. 4-20: Stopper unit with data R/W station

When the transport car detects the signal bar it slows down to its programmed minimum speed V_{low} and drives against the lifted stopper, in doing so the stop button is pressed and the motor is switched off.

In front of turnouts or in robot cells the stopper unit is always mounted with a data R/W station to determine e.g. the ID-number of the waiting transport car. The PLC checks if the transport car has to stay inside the robot cell, or if the turnout has the correct position.

The transport cars have to wait until the data communication between PLC, data R/W station and mobile data carrier is completed, and the turnout operation, depending on destination of transport car, is finished (in the case of diverting junction). In the case of a merging junction, the PLC checks the correct position of turnout, or if two transport cars are waiting, which one has right of way. If all requirements are fulfilled, the PLC unblocks the stopper, the transport car starts with minimum speed and after passing the second inductive proximity sensor the PLC blocks the stopper unit again. This immediate blocking after passing the second sensor allows the separation of several transport cars retaining with direct contact in front of a stopper unit.

4.3.3.2 Data read/write station

The data read/write station (Fig. 4-21) communicates with the mobile data carriers in the transport cars (both are part of the identification and data storage system) if they are in radio range and with the central PLC. By these means, the PLC gets the information of which transport car stands at the stopper unit respectively at the data R/W station. Depending on this information, the PLC decides the next program sequence.

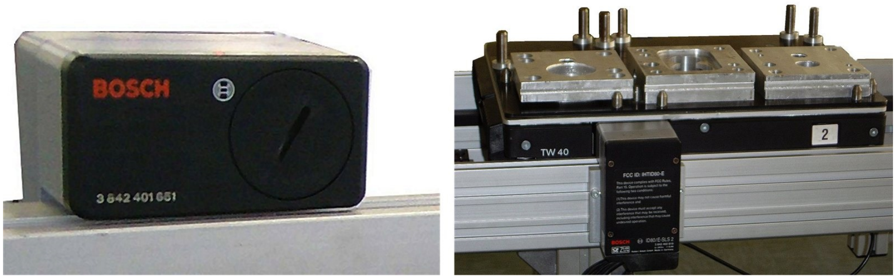


Fig. 4-21: Data R/W station front view and back view with transport car

It is possible to store additional information e.g. loading condition on the mobile data carrier. Every data R/W station communicates with the PLC using 10 binary signals (8 data bits, 2 bits for handshaking) for sending as well as receiving.

4.3.3.3 Track turnout

Without merging or diverting junctions, different routes would not be possible and thus track turnouts are important components of a transportation system like the Bosch CTS. The turnout, like the curve, basically consists of a synthetic resin base plate with plain surface, aluminium border profiles for slot-in acrylic glass for personnel safety at the outside, installed on a base frame.

Special guidance profiles are mounted on this base plate and guide the transport cars with reduced speed around the curve or straightforward through the turnout. These guidance profiles are partly moveable (yellow arrows, Fig. 4-22), and switch, controlled by the PLC, in curve position or straightforward position. The turnout shown in Fig. 4-22 is in the straightforward position.

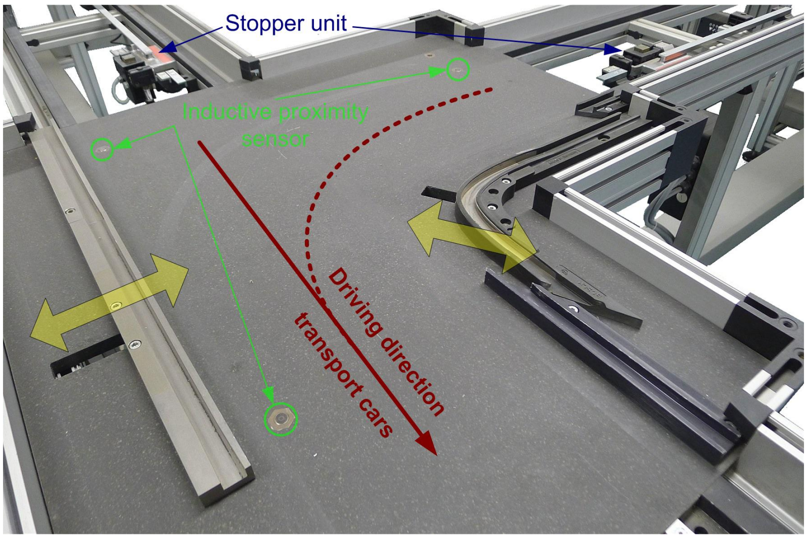


Fig. 4-22: Track turnout with merging configuration (Area 11, Fig. 4-7)

Additionally, the turnout is equipped with three inductive proximity sensors to detect transport cars entering and leaving the turnout. If a transport car is moving through the turnout, the switching operation has to be blocked by the PLC; otherwise the transport car would be jammed by the moving guidance profiles. The three inductive proximity sensors are intended for the blocking of switch operations by detecting the steel bars on the underbody of the transport car (Fig 4-12).

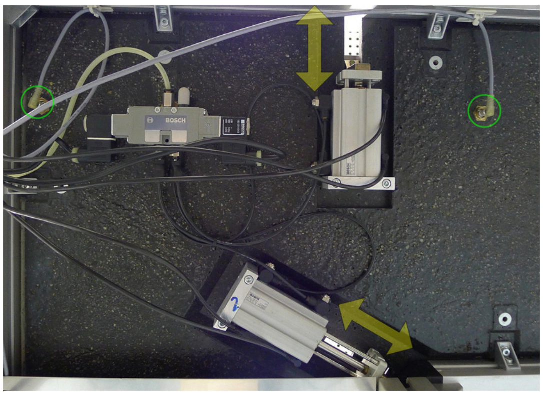


Fig. 4-23: Track turnout – bottom side view

The electrical and pneumatical components are mounted below the base plate and at the base frame. The PLC controls the guidance profiles by means of a 5/2 directional control valve and two air actuated cylinders (Fig. 4-23), each cylinder is controlled by the PLC via 1 bit. These double action cylinders provide sensors for confirming both final positions to the PLC, and allow the conclusion of turnout position or a malfunction. The CTS turnout component has four possible setup variants: diverting turnout with left or right turn and merging turnout with junction from left or right. The RFAS uses two diverting turnouts with left turn (Area 3, area 7, Fig 4-7), one merging turnout with junction from left (area 11) and one merging turnout with junction from right (area 7).

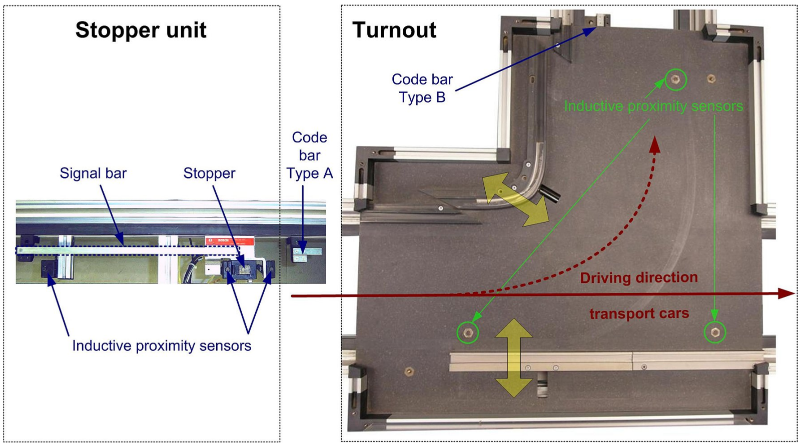


Fig. 4-24: Stopper unit and track turnout with diverting configuration (Area 3, Fig. 4-7)

Figure 4-24 shows a turnout with left turn diverting configuration in straightforward position.

4.3.3.4 Battery-charging station

The battery-charging station (charging control unit and charging adapter (Fig. 4-25/4-26) enables the constant current fast charging (U < U_{max}) and conservation charging (U ≥ U_{max}) of the built-in accumulators of the transport cars. The charging adapter is mounted together with a stopper unit, because charging is only possible when the transport car is stopped. The RFAS has battery-charging stations inside both robot cells because the transport cars have the longest stopping times there. Charging is only possible when the transport car is switched on, because the on-board microcontroller controls the charging procedure and monitors the accumulator status (current, temperature) and reports the charging via its 7-segment display.

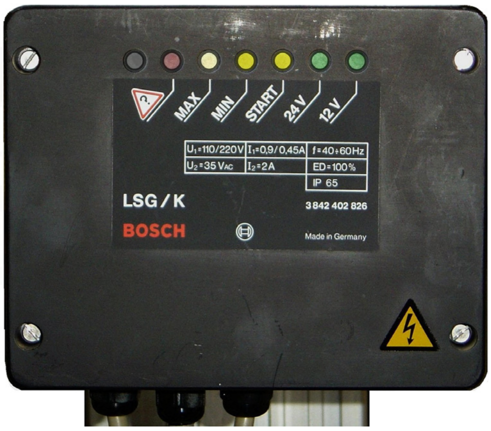


Fig. 4-25: Charging control unit

If the voltage is too low for the on-board microcontroller (e.g. deep discharge after a long period of non-use) it is not possible to use the battery-charging station, in this case a pre-charge with an external power pack is necessary. The charging control unit, being started by the PLC, monitors the accumulator voltage and shows U_{min} and U_{max} directly via LEDs (Fig. 4-25) and reports these states (or an error) to the PLC. The requirement for the transport car to drive on is reaching the minimum voltage (U_{min}), otherwise the risk of transport car breakdown before reaching a battery-charging station again exists.

4.3.3.5 Lifting-positioning unit for transport cars

The lifting-positioning unit (Fig. 4-26) is mounted together with a stopper unit, a data R/W station and a battery-charging station inside the SCARA robot cell. The entering transport car slows down and finally stops, and if it is required by the SCARA robot, the transport car will stay in the robot cell. For loading or unloading, the transport car will be lifted and fixed afterwards. The PLC controls the air actuated double action cylinder of this unit via 2 bits by means of a 5/2 directional control valve. The pneumatic cylinder provides sensors for confirming both final positions to the PLC.

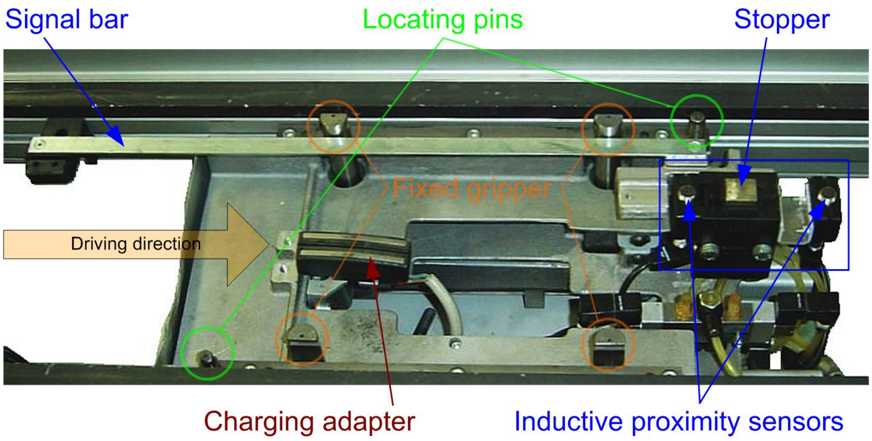


Fig. 4-26: Lifting-positioning unit inside SCARA robot cell

The lifting-positioning unit has two taper locating pins (green circles, Fig. 4-26) which are fitting into both conical bushings in the underbody of the transport car (yellow circles (2), Fig. 4-12) when they move up to fix and lift the transport car. Additionally the both steel bars on the underbody of the transport car (green rectangles, Fig. 4-12), which are now located inside the slots of the fixed grippers (orange circles, Fig. 4-26) move up against the upper edge of these slots. In this way an exact positioning and fixing (repeat accuracy ± 0,1mm) with a maximum vertical pressure of 10 kN on the load surface of the transport car is possible.

The modelling and simulation of the gantry robot cell and the SCARA robot cell is part of the continuing VC research at the UASA Hannover, but would go beyond the scope of this thesis, hence only a short description of these robot cells follows. The status of both particular projects and further planning is outlined in the recommendation for future work.

4.4 Gantry robot cell

The main part of the second robot cell is a 4- axis Berger-Lahr gantry robot (Fig. 4-27). This robot cell is used for assembling and disassembling demonstration filters.

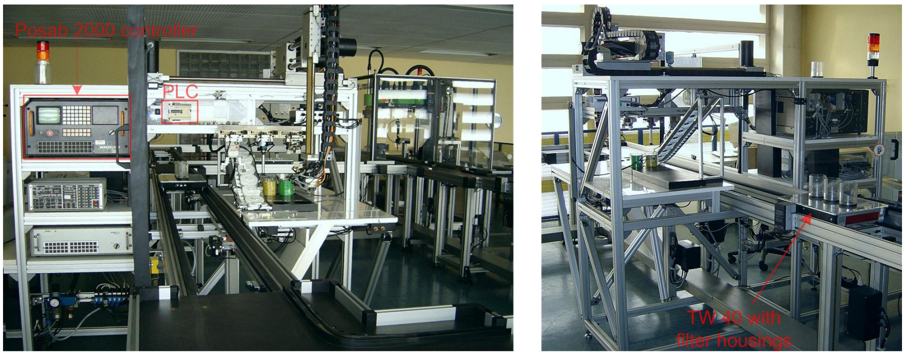


Fig. 4-27: Berger-Lahr gantry robot (front view and rear view)

The acrylic filter housings are delivered by a transport car (Fig. 4-27) and the completed filters are removed by the same type of transport car. Because this robot has the ability to disassemble the filters again, it is possible to run a circle process.

A Berger-Lahr (now Schneider Electric) TLM2 Multi Axes Motion Controller/Soft PLC with an additional Wago CANopen I/O module (Fig. 4-28) has replaced the former Berger-Lahr controller/Mitsubishi PLC combination controlling the gantry robot cell (cf. Fig. 4-27).

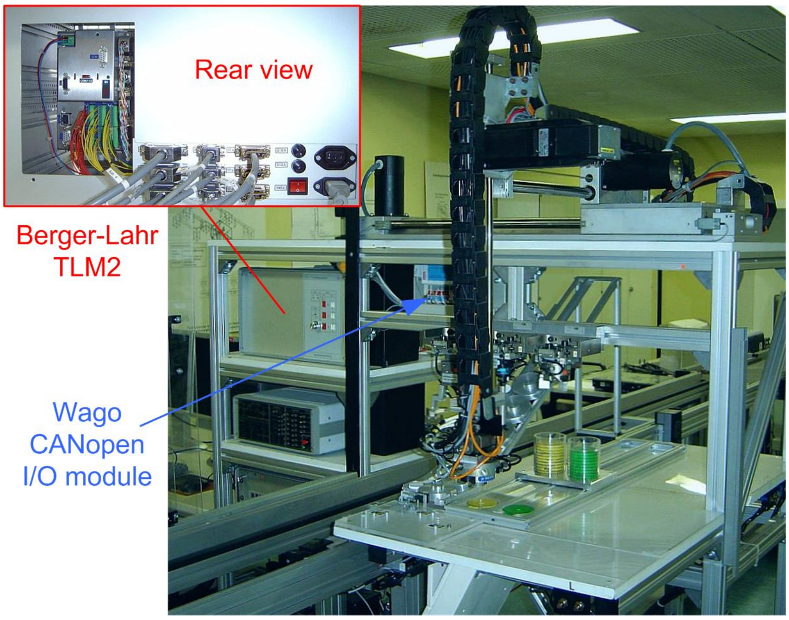


Fig. 4-28: Berger-Lahr gantry robot with new controller

The Wago CANopen I/O module is necessary because the TLM2 alone does not provide enough I/Os for the complete robot cell with gripper’s and other peripheral devices.

4.5 SCARA robot cell

The originally used SCARA robot Bosch SR800 (Fig. 4-1) has been replaced by a Mitsubishi SCARA robot. The planning of reconstruction of the SCARA robot cell has been carried out considering the technical demands (workspace needed, payload…), later CAD based off-line programming, capability for being integrated in the simulation (VC) of the complete manufacturing system and so on.

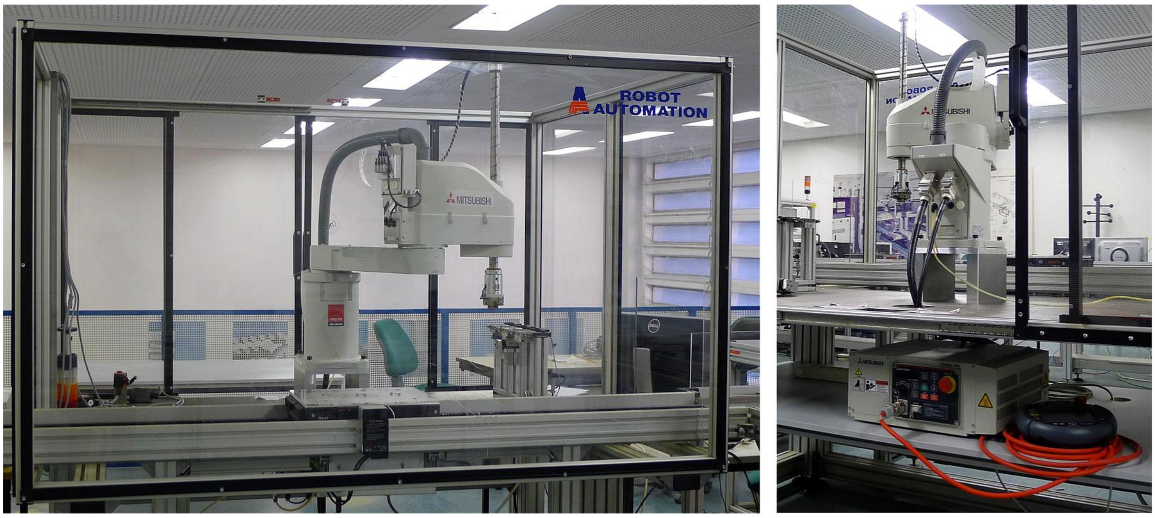


Fig. 4-29: SCARA robot cell under reconstruction with (front view and rear view)

The front view shows the Mitsubishi RH-20SDH8545 with adapter for automatic tooling system and a gripper station. Other peripheral devices are not yet in place, because operating processes have first to be defined. One process shall use linked operation together with the gantry robot. This is in contrast to the previous operational process of the Bosch SCARA robot, which had been executing a stand-alone assembly. The rear view in figure 4-29 shows the robot controller CR2D and the teach box R56TB beneath the workspace of the robot.

4.6 Specification of engineering software environment for VC research

Having established sufficiently complex hardware for the test-bed, the next task is the establishment of an appropriate software environment for the test-bed that complies with the requirements for VC (cf. subsection 5.1). The VC of the entire RFAS is a final goal of the ongoing VC research at the UASA Hannover; hence, all software tools used at the RFAS will be described below to establish the context for the software tools considered in this study.

The test-bed built shall reflect the situation at SMEs, not only for the hardware used but also for the software tools used. It is appropriate and advantageous for SMEs to further utilise their commonly used tools instead of changing the system landscape. In the case of introducing a complex integrated suite of tools from the digital factory concept, well understood and accepted tools would need to be replaced by new ones, and the user expertise, libraries, programs and other solutions would be lost (Drath et al., 2008a). In addition to the software tools for control engineering and HMI, it is necessary to select 3D CAD tools for the basic simulation model building and 3D plant simulation tool for VC.

The specification of the engineering and simulation environment for VC starts with the selection of appropriate off the shelf tools for the PLC programming, robot programming, and HMI. For the PLC and robot programming tools, the decision is linked to the choice of controller hardware. The selection of PLC hardware implies generally (with few exceptions) a commitment to a specific integrated development environment (IDE) for the engineering of control software. In the case of robots, the selection process starts with technical demands such as needed workspace, payload, performance, accuracy etc. and the choice of controller is associated with the robot. If robots from different vendors are under consideration the choice may be made for reasons of the IDE utilised and associated robot language (besides e.g. monetary aspects). This kind of selection may be done because the software is already used in the company, due to customer wishes etc. or as in the case of the SCARA robot of the RFAS, simulation options available for VC. For PLCs it is possible to choose a hardware independent IDE such as Codesys (see below) first, and select a fitting controller hardware from different vendors afterwards.

4.6.1 PLC programming – STEP7

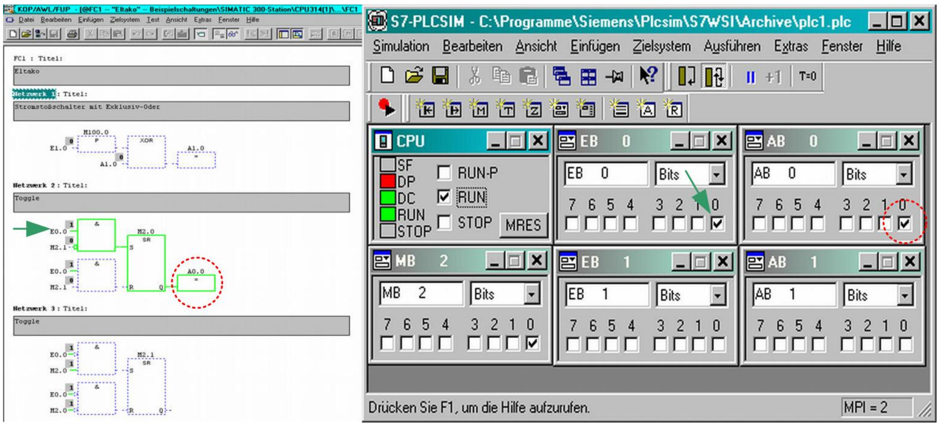
The new Siemens S7 PLC system has been selected for the RFAS, amongst other reasons because of the wide distribution of Siemens S7 PLCs and its associated STEP7 IDE in Germany. In 2007 Siemens referred to a study of the ARC Advisory Group and stated a world market share for PLCs of 28,7%, followed by that of Rockwell Automation (Allen-Bradley) with 21,8% and Mitsubishi with 14,9% (Siemens, 2007). According to (Hönle, 2014) Siemens S7 has to be considered as the market leader for PLCs in Europe. Because of this leading market position many other software tools, and also plant simulation tools (cf. chapter 2) provide interfaces for direct coupling/data exchange with S7 systems.

All controller programs for the RFAS developed in the future, either for PLC or robot controllers, shall be verified by conducting a VC. STEP7 is the Siemens IEC 61131 software for the configuration and programming of the SIMATIC S7-300 and S7-400 automation systems, and STEP7 V 5.x is used as programming system for the new central PLC with CPU S7-317-2 PN/DP. It provides the standard programming languages defined in part 3 of the IEC 61131 (IEC 61131-3, 2003).

IEC 61131-3		Siemens STEP7	
Instruction List	IL	Statement List	STL
Ladder Diagram	LD	Ladder Diagram	LAD
Function Block Diagram	FBD	Function Block Diagram	FBD
Structured Text	ST	Structured Control Language	S7-SCL
Sequential Function Chart	SFC	Sequence Control Editor	S7-GRAPH
		Continuous Function Chart	S7-CFC
		State Graph Editor	S7-HiGraph

Table 4-2: Comparison of IEC 61131-3 and STEP7 programming languages

Statement List, Ladder Diagram and Function Block Diagram tools are supplied as standard by Siemens, while S7-SCL and S7-GRAPH are optional engineering tools. The additional Continuous Function Chart (CFC) and the state graph language S7-HiGraph are not part of the IEC standard. The tool S7-PLCSIM, supplied as standard, emulates the controller hardware and allows the functional testing of PLC programs (Berenbach et al., 2003), but without process simulation this is of limited suitability, useful for very small programs or single functions in a program (Fig. 4-30). The necessity for a process model to undergo functional verification has already been outlined in (Litz et al., 1998).



KOP/AWL/FUP - [@FC1 -- "Eltako" -- Beispielschaltungen\SIMATIC 300-Station\CPU314(1)\...\FC1

FC1 : Titel:
Eltako

Netzwerk 1: Titel:
Stromstoßschalter mit Exklusiv-Oder

Netzwerk 2 : Titel:
Toggle

Netzwerk 3 : Titel:
Toggle

S7-PLCSIM - C:\Programme\Siemens\Plcsim\S7WSI\Archive\plc1.plc

Drücken Sie F1, um die Hilfe aufzurufen.

MPI = 2

Fig. 4-30: PLCSIM and STEP7 (FBD) online

Besides its limited use for functional verification, this virtual controller can be coupled with simulation tools and can facilitate a VC with simulated plant and emulated control system (SIL simulation), which has been projected as one possible partial solution for realising a VC of the transportation system.

The collegiate controller programs for the central S7 PLC are developed henceforward by means of STEP7. The basic design requirement for the current program version has been the separation of the PLC code for the control of hardware elements from the code for the operating sequences. For the components of the transportation system presented above (e.g. turnout, stopper unit,…), corresponding software components in form of STEP7 function blocks as described in (Berenbach et al., 2003), have been developed. Future control programs are to be developed based on these reusable library elements. This modularisation of PLC programs (using functions or function blocks related to hardware components) supports mechatronic engineering, and is partly used in industry. These software modules can be tested individually and allow a better re-use for new projects than copy and paste code from old project to new projects.

Such a component-based approach necessitates the decomposition of automated manufacturing systems into configurable stand-alone mechatronic components, sub-systems and associated control functions that can be then tested separately. The RFAS has been decomposed into transportation system and two robot cells. The transportation system has been decomposed into hardware components that are partly predetermined by the vendor catalogue. These components have been equipped with associated control functions in form of STEP7 function blocks where applicable.

Besides manual operation (switching e.g. stopper units and turnouts via operator touch panel), an automatic mode and a test program are selectable in the current STEP7 program. The automatic mode allows the operation of 1-4 transport cars in any order and with selectable assignment to the robots. The test program is used to check all sensors and actuators by one transport car driving on all tracks of the complete system, which is realised with a SFC (S7-Graph). This PLC program is the first one to be verified with a VC of the transportation system, and thus the test case for the designed VC environment.

4.6.2 HMI programming – WinCC Flexible

A Siemens operator panel has been selected to operate the transportation system. For this reason the HMI engineering has to be done with the associated Siemens software WinCC Flexible 2008, currently the previous version SP3 is installed. Several graphical control displays for operation and visualisation of the current PLC program (manual mode, automatic mode and test program) mentioned above have been developed for the RFAS. The Siemens tools STEP 7 and WinCC Flexible are considered in this study for the VC of the transportation system, but it is also a project target to show that it is possible to provide the facility to integrate the robot programming for both robot cells of the RFAS to the developed VC environment.

4.6.3 Gantry robot programming – CoDeSys

The **Co**ntroller **De**velopment **Sys**tem - CoDeSys (3S, 2014) is the IEC 61131 IDE for controller programming of 3S-Smart Software Solutions. It is currently available in two versions. CoDeSys V 2.3 offers like STEP7 all languages defined in the IEC standard and a CFC editor is included by default.

In contrast to STEP7 it is hardware independent, which means an engineer is able to program controller hardware from many different manufacturers (e.g. ABB, EATON, FESTO and WAGO) with the same programming system. This is realized by integrating target support packages from the controller vendor and compiling the code for this target hardware. CoDeSys V2.3 is also the programming system for the Berger-Lahr TLM2 motion controller of the gantry robot, complemented with target support package and motion controller libraries provided by Berger-Lahr (now Schneider Electric). The newer CODESYS V3.5 extends the IDE for example with object-oriented programming (OOP). This version is to date not as widely supported with controller runtime versions as V2.3, but some new PLCs compatible with V3 are available.

In contrast to STEP 7 V5.x, CoDeSys provides a built-in basic visualisation (cf. Fig. 6-16). This is an advantage because the HMI can be developed with the same tool as the controller application, and it allows a more intuitive functional verification of small programs than the display with bits and bytes by S7-PLCSIM. An OPC-Server is included in both versions by default, and allows the standardised coupling with many other tools such as SCADA/HMI tools or crucial for VC, the coupling with plant simulation tools like e.g. CIROS.

4.6.4 SCARA robot programming – Melfa Basic IV

Besides the on-line programming by teach-in procedure, using the teach box R56TB (Fig. 4-28), the Mitsubishi software RT ToolBox2 allows textual off-line programming. The additional software tool Mitsubishi Melfa-Works, an add-in to the CAD tool SolidWorks, provides CAD based off-line programming. Furthermore, Mitsubishi robots and their programming language Melfa Basic are supported by CIROS. By means of Melfa-Works and alternative CIROS, it is possible to conduct a VC of the robot cell, whereas only CIROS provides the extra option of a VC in combination with the transportation system. This is not possible with Melfa-Works. Both cases require a previous measurement and 3D CAD design of the robot cell during future work. A 3D model of the RH-20SDH8545 is expectedly available for Melfa-Works not only, but also for CIROS, though the peripheral devices have to be modelled with a 3D CAD tool.

4.6.5 3D CAD-Tools

Besides the tools for control engineering and HMI, it is necessary to use 3D CAD tools, because 3D CAD is necessary to build the virtual manufacturing system model. The CAD preparation and development of the CAD library containing the components and sub-systems of the transportation system has to be done with 3D CAD tools. The UASA Hannover has licences of widely used tools available, and the following choice has been used: Siemens PLM (former UGS) Solid Edge, Dassault SolidWorks and Pro/Engineer (PTC Creo Parametric).

5 Requirements for VC and simulation model building

Amongst other elements, software tools as mentioned in subsection 4.6 are necessary in for example the design and operation of manufacturing systems, and are thus commonly used. However, this does not apply to simulation tools. Besides the requirements for VC mentioned below, there is a need for qualified personnel to carry out a VC project. Because of the different engineering disciplines involved, it is not sufficient to have focused automation engineering skills; additional knowledge about the options and limitations of 3D CAD and especially 3D plant simulation tools is required. This implies a challenge for control engineers in industry wanting to conduct a VC without appropriate support. A simulation expert is not necessarily available in the company, which is particularly the case in SMEs (cf. chapter 2).

5.1 Requirements for VC

According to Makris et al. (2012) the realisation of a VC project has different "data requirements". Makris and co-authors do not only specify data items in their list, but also e.g. the hardware control systems. Hence, in this thesis the requirements have been split into data requirements and hardware and software tool requirements. Besides, some missing items such as unmodified controller programs, control displays for HMI/SCADA and 3D plant simulation system, additional information related to the topics listed has been presented.

5.1.1 Data requirements

- 3D simulation model of the manufacturing system to be commissioned

  This virtual plant model comprises mechatronic models of components and sub-systems containing geometry, kinematics, electrics and if applicable logics describing the behaviour. The implemented actuator/sensor functions need parameterisation, thus all thereto necessary technical data such as stroke and speed of moving actuators or physical principle and measurement range of sensors must be available, depending on planned LOD.

- Detailed layout of the manufacturing system

  The layout allows the exact placement and interconnection of components and sub-systems in the virtual plant model.

- Information about the material flow

  This comprises the sequences of operations and the interrelationship between the manufacturing processes involved, respectively facilities such as different work stations (e.g. robot cells for assembly processes).

- Controller programs for PLCs and robots

  It is necessary to use the real, ideally unmodified, controller programs for a VC (Drath et al., 2008a), otherwise the exact reproducibility of control behaviour cannot be guaranteed (Makris et al., 2012). This requirement is related to the use of real hardware controllers or virtual controllers (see below).

- Detailed lists of I/O signals

  The I/O lists are necessary to specify the mapping between actuators and sensors of (virtual) manufacturing system components and the controller I/O's.

- Control displays for HMI/SCADA systems

  Because current manufacturing systems are generally operated by means of operator panels or PC based HMI/SCADA systems, the developed control displays (possibly containing additional control functionality in form of logic and scripts), they must be included as part of the VC, unchanged as well.

- Definition of additional functionalities and signals e.g. emergency stop switches in different safety circuits or alarm messages that should also be considered in the VC.

With the exception of the 3D simulation model all these data are generated during standard procedures during the engineering of the manufacturing system.

5.1.2 Hardware and software tool requirements

- 3D plant simulation system

  The 3D simulation of the manufacturing system can be conducted either using an all-in-one simulation tool such as CIROS, or, implemented e.g. as by Makris and co-authors, using a co-simulation of two simulator tools for separate simulation of a 3D model (INVISION in this case) and behavioural model (WinMOD in this case).

- Real hardware controller or virtual controller

  A VC can be conducted using HIL and/or SIL configurations. In the case of HIL simulation the real hardware controller (PLC / robot) are already necessary at the time of VC. For SIL simulation a VC requires virtual controllers, able to execute the original programs, for all implemented PLCs and robots. This means, the vendor specific native robot programming language (e.g. ABB Rapid, KUKA KRL, Mitsubishi Melfa Basic) or PLC code (e.g. STEP7), should be available and applicable. Makris et al. emphasise a general problem of SIL, the *"low availability of up-to-date control simulation packages for a particular control version"*. Current plant simulation tools provide only a limited choice of virtual controllers, and the selection of which vendor specific controller code is supported differs from one simulation tool to another (Bockstette, 2013). Due to this disadvantageous situation, engineers planning to set-up a VC have to check the functionality of plant simulation tools also regarding their ability to execute the native code of controllers implemented in the real manufacturing system internally, to couple external virtual controllers (both SIL) or to couple real hardware controllers (HIL). The coupling of external controllers is as limited and tool dependent as the simulation tools' ability to execute native code internally.

- Hardware or software communication infrastructure
  The coupling of external controllers, either real or virtual, with the virtual plant model within the simulation system, requires a communication infrastructure. The direct coupling with PLCs in HIL simulation is possible via e.g. Ethernet/IP, fieldbusses or OPC. If a PLC specific protocol is used, the simulation tool has to support the particular PLC with a specific driver. The use of the OPC standard is widely supported. In this case, the simulation tool has to provide an OPC client that communicates with the OPC server of the PLC system. The communication via OPC comes into consideration if no fieldbus is used or its time response is negligible and no safety signals, part of VC, are transmitted by fieldbus. In the case where the simulator PC is equipped with a fieldbus card for direct coupling via fieldbus, the simulation tool has to provide a driver interface for this fieldbus card. Another possibility is to emulate the fieldbus using special hardware devices (e.g. Siemens SIMBA); in this case, a driver is needed for the simulation tool to support such an emulator device. Currently, only few tools support direct coupling via fieldbus or fieldbus emulation. For SIL simulation, it could be possible in some cases to couple the OPC client to a SoftPLC via an OPC server, but in the most cases, a specific driver for the particular virtual PLC would be necessary.
- In the case of HIL simulation, it is possible to couple the real operator panel or PC based HMI / SCADA system with the hardware PLC, as in real operation of the manufacturing system. If a SIL simulation is to be conducted, a virtual controller has to be coupled with e.g. the PC runtime version of an operator panel HMI or a PC based SCADA system, which implies generally special drivers or protocols.

Besides such technical requirements (data, HW, SW), it is also important to clarify some principal and organisational issues to be addressed by a company intending to implement VC. According to the first two steps of the proposed procedure for a VC simulation study (Fig. 3-1), the following exemplar questions need to be answered:

- What is the aim of the VC? Why should a VC be conducted? Which functionalities of the real systems should be tested? Is the focus of VC only on the V&V of automation systems or are some details of the technical process also important? The answers to these questions will lead to a requirement specification that will indicate how the simulation model building should proceed, (i.e. what has to be modelled and the level of detail and functionality that is required).

- Which staff members (e.g. plant layout designer, mechanical/electrical CAD designer, PLC programmer, HMI designer, commissioning engineer, IT specialist, management etc.) should be involved in the implementation of VC, and which roles should they take on?
- Who will execute the tasks within the steps of figure 3-1? Particularly, who will do the modelling? Is it possible to train existing staff member(s)? Is recruitment of a simulation specialist necessary, or is subcontracted modelling by a service provider a better way?
- How should the results of VC be documented?

5.2 Specification of VC environment

For VC it would in principle be possible to use relevant tools, such as Delmia Automation and Siemens Process Simulate & Commissioning, which are both part of the market-leading off-the-shelf factory planning suites of tools used in e.g. the automotive industry. Actually, SMEs often consider these suites of tools as too complex to use and to reasonably incorporate in their in-company engineering workflows (Hoffmann et al., 2012). The *"high learning threshold"* of Delmia compared to simulation tools like e.g. Visual Components 3D-Create has also been a complaint in (Hollander & Sappei, 2011). Additionally, criteria such as high costs for e.g. licences prevent the appreciable use of these tools in SMEs (cf. chapter 2). Hence, a less complex tool for VC, which is more easily utilised and hence likely to better suit SMEs too, will be used in this thesis. As far as possible, this should be an all-in-one simulation tool, not a co-simulation with two simulator tools.

The selection of a suitable 3D plant simulation tool belongs to procedural step 2 (Project Planning) of the proposed methodology for a VC simulation study (Fig. 3-1). The tool to be selected here must have the ability to simulate the complete manufacturing system (RFAS), not only the transportation system. The VC of the transportation system (STEP7 program in conjunction with WinCC Flexible HMI) is considered in this study, but the final goal is the VC of the entire RFAS, a simultaneous simulation of the transportation system in cooperation with the gantry robot cell (CoDeSys) and the SCARA robot cell (Melfa Basic IV). This VC should be able to detect control code errors as well as errors of the physical setup or rather errors resulting from interactions between different objects e.g. the transport cars, their payload, and the robots. Because current plant simulation tools provide only a limited choice of internal virtual controllers or coupling possibilities, the functionality of available plant simulation tools has to be checked regarding these demands.

Based on the general requirements for VC (see above) and the particular hardware of the RFAS the following essential requirements for the VC environment and the selection of a 3D plant simulation tool have been defined:

- Applicability of the original, unmodified controller programs for S7 PLC and robot controller (CoDeSys and Melfa Basic) either by coupling to an external real or virtual controller or by execution of original code by internal virtual controller
- Applicability of control displays for operator touch panel (WinCC Flexible)
- Possibility to simulate different types of sensors and wireless data communication
  The tracks of the transportation system use e.g. inductive proximity sensors, and the transport car additionally e.g. optical distance sensors. The data R/W stations exchange data with the mobile data carriers in the transport car by means of wireless communication (cf. Subsection 4.3).
- Capability to build and use 3D simulation models with kinematics (cf. Fig. 2-5).
- Extendible library for simulation models
  The library has to be extendible to store the simulation models of components and sub-system of the transportation system. The Mitsubishi SARA robot implemented in the RFAS should preferably be represented in this library with a ready to use 3D simulation model.
- Possibility to import 3D CAD data with standardised exchange CAD formats and native CAD formats if applicable
  The import of CAD data is necessary to build new simulation models. The support of standard exchange formats (e.g. STEP) is favourably because the choice of CAD tool is not limited to supported native formats then.

The simulation of electrical and pneumatic equipment and wiring has been considered by the author to be dispensable for the VC of the RFAS; similarly, the integration of communications infrastructure such as field bus systems by simulation. Currently the RFAS does not make use of a fieldbus system for the communication between central PLC and robot controllers, and for the projected Profibus-DP, the time response delays have been considered to be negligible for the reported simulations. This limitation of simulation scope might be considered differently in other applications.

In the early phase of this research project, the 3D plant simulation tool COSIMIR (**C**ell **O**riented **Sim**ulation of **I**ndustrial **R**obots) was identified as a promising candidate for the VC of the RFAS. Preliminary analyses of its features (see below), usability and initial modelling tests of e.g. the track turnout indicate its appropriateness in principle. A later repeated updated survey of plant simulation tools (cf. subsection 2.8) substantiated this appraisal and again rated this tool as one of the best suitable for the VC of the RFAS (Bockstette, 2013).

5.2.1 3D plant simulation tool

The industrial 3D simulation system CIROS (former COSIMIR) has been selected as plant and controller simulation tool for VC. COSIMIR has been sold for several years as a commercial tool by the RIF e.V. (RIF, 2009) before it was renamed to CIROS (**C**omputer **I**ntegrated **Ro**bot **S**imulation). It has been originally developed in the 1990s (Freund et al., 1993) as *"user friendly"*, *"intuitively usable"* and *"highly modular"* robot simulation system (Freund et al., 1994). An overview of its internal structure is presented in (Freund & Rossmann, 1995). Originally intended for off-line programming and simulation of robot-cells, the concept covers from the beginning the interaction of robots with additional devices such as grippers, conveyor-belts or sensors. The basic concept is designed such generally that it allows up to now the integration of new methods or features such as PLC simulation, and by further development the tool is now suitable for the simulation of complete manufacturing systems (Rossmann et al., 2010).

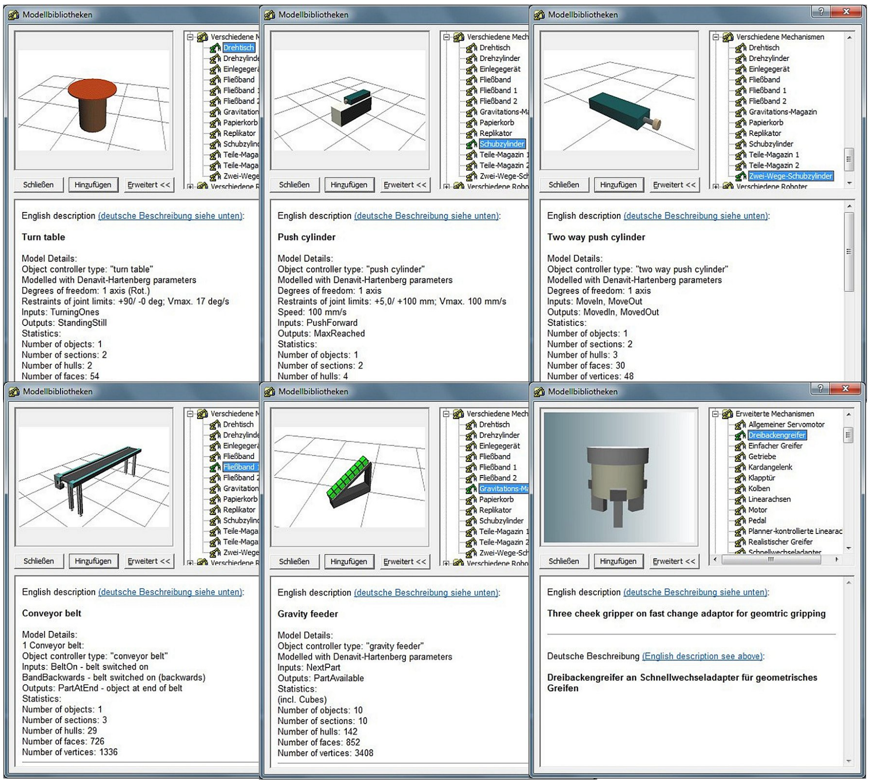
The industrial simulation system CIROS comprises the modular core system CIROS Studio with its optional extension modules, and the additional tools CIROS Planner, CIROS Programming and CIROS Production/Supervision (RIF, 2012). The most important features of CIROS Studio, subsequently referred to as CIROS, are presented below. CIROS Planner allows production process design by e.g. cycle time planning with automatically created PLC programs as SFCs from the cycle time diagrams (Gantt charts). CIROS Programming is a stand-alone IDE for Mitsubishi robots supporting off-line programming with MELFA BASIC III/IV/V and MOVEMASTER COMMAND as well as on-line functions (e.g. teach-in, monitoring,…), all features are also included in CIROS Studio. CIROS Production/Supervision, the further development of COSIMIR Factory (Freund & Pensky, 2002), can be used to create an elementary version of a Manufacturing Executing System (MES) for production lines.

Educational licences are distributed by FESTO Didactic (FESTO, 2015a). The CIROS Automation Suite comprises all educational tools including CIROS Studio. Additionally FESTO distributes versions that only allow the simulation of previously modelled educational mechatronic system (FESTO MPS) with CIROS Mechatronics or predefined robot work cells with CIROS Robotics.

The applicability of CIROS to this research project and accordingly the VC of the RFAS results from following points, which are partly achieved through its extension modules, and are included in the educational licences used at the UASA Hannover:

- Library of predefined or self-created mechatronic simulation objects
  CIROS provides an internal library with several mechatronic components, robot models from different vendors, predefined functional groups or sub-systems such as conveyor systems, assembly

stations, handling stations and stocks (e.g. from FESTO FMS) and other hardware components. Furthermore the library contains basic mechanisms for e.g. rotation (turntable, motor), translation (one-way/two-way push cylinder), conveyor, gravity magazines, grippers, switches, replicator, trash bin etc. as well as extended mechanisms such as three cheeks gripper, generic servo drive etc., either with geometry (Fig. 5-1) or as invisible mechanism.



English description (deutsche Beschreibung siehe unten):

Turn table

Model Details:
Object controller type: "turn table"
Modelled with Denavit-Hartenberg parameters
Degrees of freedom: 1 axis (Rot.)
Restraints of joint limits: +90/ -0 deg; Vmax. 17 deg/s
Inputs: TurningOnes
Outputs: StandingStill
Statistics:
Number of objects: 1
Number of sections: 2
Number of hulls: 2
Number of faces: 54

English description (deutsche Beschreibung siehe unten):

Push cylinder

Model Details:
Object controller type: "push cylinder"
Modelled with Denavit-Hartenberg parameters
Degrees of freedom: 1 axis
Restraints of joint limits: +5,0/ +100 mm; Vmax. 100 mm/s
Speed: 100 mm/s
Inputs: PushForward
Outputs: MaxReached
Statistics:
Number of objects: 1
Number of sections: 2
Number of hulls: 4

English description (deutsche Beschreibung siehe unten):

Two way push cylinder

Model Details:
Object controller type: "two way push cylinder"
Modelled with Denavit-Hartenberg parameters
Degrees of freedom: 1 axis
Inputs: MoveIn, MoveOut
Outputs: MovedIn, MovedOut
Statistics:
Number of objects: 1
Number of sections: 2
Number of hulls: 3
Number of faces: 30
Number of vertices: 48

English description (deutsche Beschreibung siehe unten):

Conveyor belt

Model Details:
1 Conveyor belt:
Object controller type: "conveyor belt"
Inputs: BeltOn - belt switched on
BandBackwards - belt switched on (backwards)
Outputs: PartAtEnd - object at end of belt
Statistics:
Number of objects: 1
Number of sections: 3
Number of hulls: 29
Number of faces: 726
Number of vertices: 1336

English description (deutsche Beschreibung siehe unten):

Gravity feeder

Model Details:
Object controller type: "gravity feeder"
Modelled with Denavit-Hartenberg parameters
Inputs: NextPart
Outputs: PartAvailable
Statistics:
(incl. Cubes)
Number of objects: 10
Number of sections: 10
Number of hulls: 142
Number of faces: 852
Number of vertices: 3408

English description (deutsche Beschreibung siehe unten):

Three cheek gripper on fast change adaptor for geomtric gripping

Deutsche Beschreibung (English description see above):

Dreibackengreifer an Schnellwechseladapter für geometrisches Greifen

Fig. 5-1: Exemplary CIROS mechanisms included in library

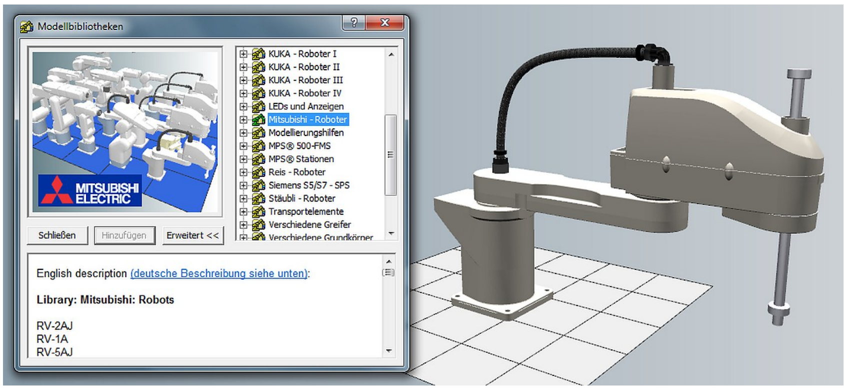
The behaviour modelling for mechatronic components in CIROS is governed by these mechanisms; they are not changeable by a user, but can be combined and parameterised. It is also possible to add behaviour by implementing a logic controller into a component. A basic concept, implementable with several 3D simulation tools including CIROS is that of two-level modelling. High-level modelling can be considered as the aggregation of simulation models from such a simulation model library, into a virtual manufacturing system. If appropriate component models for the intended simulation are not available in this library, a complex and time-consuming low-level modelling exercise to build new mechatronic models for the library involving the above mechanisms, becomes necessary.

- PLC simulation

  CIROS allows two kinds of SIL simulation for S7 PLCs. It is possible to import original Siemens STEP7 programs directly into CIROS and to execute them with the internal virtual controller, or to couple CIROS to S7-PLCSIM and use this external PLC emulator instead. Other PLCs can be utilised by coupling via OPC (see below) in a HIL simulation.

- Robot simulation

  Different robot manufacturers are supported by ready to use 3D models; currently robots from e.g. ABB, Adept, DENSO, EPSON, FANUC, KUKA and Mitsubishi (Fig. 5-2) are in the library.



English description (deutsche Beschreibung siehe unten):

Library: Mitsubishi: Robots

RV-2AJ
RV-1A
RV-5AJ

Fig. 5-2: Mitsubishi robots in CIROS library

  By default, virtual controllers in CIROS, also for robots, are programmed with the neutral robot programming language IRL (Industrial Robot Language). Additionally CIROS supports some native programming languages (including e.g. ABB Rapid, KUKA KRL and Melfa Basic IV required for the Mitsubishi SCARA robot at the RFAS). These robot programs will be translated by a compiler framework (Freund et al., 2001a) in a standardized IRDATA (Industrial Robot Data) program, which is interpreted by the integrated robot controller PCROB (Freund et al., 1993). This controller executes the kinematic calculations and controls the simulated robots, or where applicable the real robots (Rossmann et al., 2010).

- Multi controller simulation

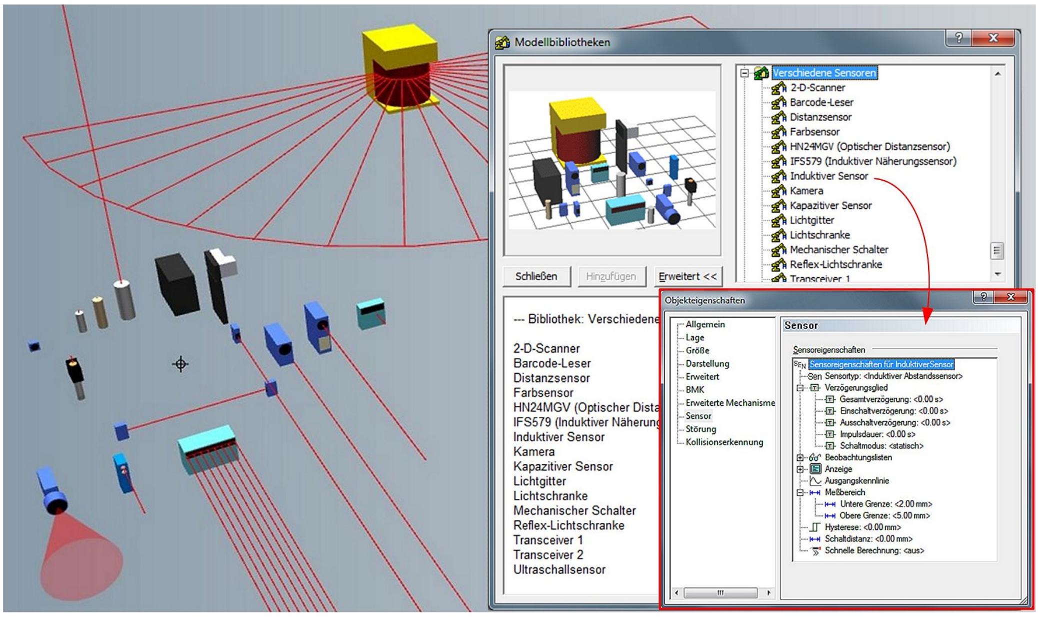
  It is possible to apply several internal virtual controllers (PLC and robot controller) and couple external controllers (real or virtual) at the same time, this is necessary for the VC of a manufacturing system with cooperating controllers as that one at the UASA Hannover (Fig. 5-5).

- OPC Client / Server functionality

  CIROS provides an OPC client, this allows for the coupling of CIROS with e.g. the CoDeSys OPC server. Thus, a HIL simulation with the Berger-Lahr TLM2 motion controller of the gantry robot is possible. This would also allow a HIL simulation with hardware PLCs if a fitting OPC server were provided.

- Sensor simulation

  Sensor models comprises different physical measurement principles such as capacitive, inductive, optical or ultrasonic producing analogue or binary output signals, possibly pulsed, delayed or with hysteresis. The utilisation of these various methods is only possible if the objects to be detected by these sensors provide material properties such as reflection (optical and acoustical), permeability, permittivity and conductivity (Freund et al., 1994). By this means many different kinds of real sensors such as inductive proximity sensors (like in tracks and transport cars of the RFAS) or optical sensors, for example light barrier, light grid, 2D laser scanner, colour sensor, barcode reader, camera and optical distance sensors (like in the transport car) can be simulated.



Sensoreigenschaften für InduktiverSensor
Sensortyp: <Induktiver Abstandssensor>
Verzögerungsglied
Gesamtverzögerung: <0.00 s>
Einschaltverzögerung: <0.00 s>
Ausschaltverzögerung: <0.00 s>
Impulsdauer: <0.00 s>
Schaltmodus: <statisch>
Beobachtungslisten
Anzeige
Ausgangskennlinie
Meßbereich
Untere Grenze: <2.00 mm>
Obere Grenze: <5.00 mm>
Hysterese: <0.00 mm>
Schaltdistanz: <0.00 mm>
Schnelle Berechnung: <aus>

Fig. 5-3: CIROS sensors

These sensors are provided in the model library. They can be used with given geometry (Fig. 5-3), but also the sensor mechanisms are only stored in the library, to be assigned to imported geometries. The sensor parameters (such as switch-on/switch-off delay, pulse duration, measurement range, hysteresis, characteristic curve for analogue values) are adjustable, the measurement range is especially important for the simulation of the transportation system tracks and transport cars. The object property window of an inductive sensor is shown in figure 5-3 as an example, highlighted by a red rectangle. In order to limit the necessary computing power, the sensor's measuring range is approximated by a line, a circle-sector shaped or rectangle shaped fan of lines or a cone (Fig. 5-3), and their density is also parametrizable (Rossmann et al., 2010). If the simulated sensor detects an object, the colour of lines changes and assists the user in visual analysis.

CIROS provides the largest variety of sensors and measurement principles amongst all simulation tools reviewed in (Bockstette, 2013). Today the parameters have to be manually assigned to the sensors in the simulation models as do the material properties.

- Simulation of transceiver/transponder
  This feature is necessary for the simulation of communication between the transport cars with their mobile data carrier communicating and the data R/W stations.

- Import filter / Export filter for CAD data

  The currently supported import formats are the 3D exchange formats STEP (AP203 / AP214b), IGES (5.0), VRML (1.0 und 97) and STL (Table 5-1).

	Geometry	Structure	Colour	Material	Textures	Kinematics
STEP	Y	Y	Y	conditionally	N	N
IGES	Y	Y	Y	conditionally	N	N
VRML	Y	Y	Y	Y	Y	N
STL	Y	N	N	N	N	N

Table 5-1: **3D CAD import formats CIROS (based on CIROS online help)**

  The main drawback of supported CAD formats and filters is the impossibility to import the kinematics from 3D CAD tools. Since 2012 CIROS supports the import of native models from Autodesk Inventor (RIF, 2013), but kinematics transfer is neither supported.

- Transport Simulation for track-bound transportation systems

  The simulation of track-bound transportation systems can be simplified, if applicable, by a plug-in providing a special modelling method called "Transport Simulation". The passive carriers are equipped with an "anchor" and move along active track "segments" between "nodes", such a "node" would be for example a stopper (Rossmann et al., 2010). "Connections" of segments are used to build the layout of transportation system, supported by a snapping functionality. This modelling method is able to consider some physical characteristics such as gravity, friction coefficients for the modelling of the frictional grip between tracks and carrier and speed ramps with constant acceleration of drives (Wischnewski & Freund, 2004). These physical effects are only implemented in the Transport Simulation in a proprietary way, and are not usable in CIROS without this plug-in, because CIROS (up to version 5 currently used at the UASA Hannover) does not make use of a general physics engine. The quite recently introduced version 6 provides the general availability of physical effects by implementing Nvidia PhysX (NVIDIA, 2016).

Additionally CIROS provides:

- Malfunction simulation

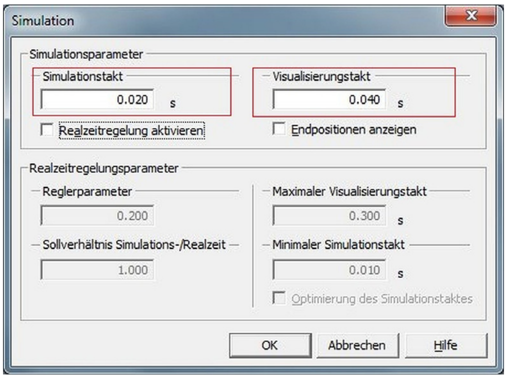
  The testing of different failure scenarios (e.g. sensor defects, broken actuators) is possible (Rossmann & Heinze, 2010), and supported by a GUI to set failures or to define test sequences. If conducted during VC, this would allow checking for adequate response of control programs to failures which occur, without endangering people or the risk of damages. Besides, operators can be trained to identify system maloperation and to handle emerging problems. Maintenance personnel can be trained in failure troubleshooting. For students in automation labs it is likewise useable.

- Collision detection

  Such a feature is useful to verify robot programs regarding the mechanical setup of the robot cell as well as together with transport cars and their payload inside the robot cell during a VC. It is possible to indicate a collision by changing the colour of colliding objects, or by stopping the simulation. The choice of objects to be included in the collision detection has to be well considered because typically, it will not be possible to choose a large number due to high computational overheads. Alternatively, it is expedient to use a simple enveloping geometry, e.g. a boundary box, instead of the complex geometry for collision calculation.

- Simulation calculation independent of 3D graphics calculation

  CIROS provide configurable visualisation rates that are independent of the calculation cycle of simulation (geometry and function/behaviour separated).



Simulationstakt: 0.020 s

Visualisierungstakt: 0.040 s

☐ Realzeitregelung aktivieren

☐ Endpositionen anzeigen

Reglerparameter: 0.200

Maximaler Visualisierungstakt: 0.300 s

Sollverhältnis Simulations-/Realzeit: 1.000

Minimaler Simulationstakt: 0.010 s

☐ Optimierung des Simulationstaktes

Fig. 5-4: Simulation and visualisation cycle parameters

  The minimum cycle time for simulation and visualisation is 0,001s; the time for visualisation has to be greater or equal to the simulation cycle time. An automatic control of parameters depending on the available calculation power of the computer is possible.

- XML model interface

  Currently only the system internal model exchange via XML from CIROS Production Simulation to CIROS Production Supervision is supported. Thus, the integration of CIROS with other tools supporting XML e.g. via AutomationML is *"still complicated, because CIROS environment does not support strongly XML data formats"*, which has been also detected in (Müür & Pettai, 2010).

New releases of CIROS are generally provided with new visualisation features such as shadows and enhanced surface reflections by lights. On the one hand, because increased calculation power permits this, on the other hand presumably because a realistic look for a virtual manufacturing systems can be important for the credibility of a 3D simulation, at least to persuade non-specialists in simulation such as a decision maker from management.

Besides these characteristics, the CIROS simulation system provides semi-immersive and fully-immersive VR (cf. subsection 2.7.2). Immersion through the use of VR environments (RIF, 2014), especially fully-immersive VR by using e.g. a CAVE or HMD and input devices such as data gloves would allow a VC with immersive user interaction in the 3D model during simulation. While simulating a transportation system, it would e.g. be possible to take down a virtual transport car from the virtual tracks to test the PLC reaction in VC (Rossmann et al., 2010). Immersive user interaction would also be imaginable as part of the simulation of a manufacturing system with partial manual operation (handwork place) instead of programming a virtual human.

5.2.2 Novel application of CIROS to a transportation system with self-driving carriers

To the author's knowledge and on enquiry to a developer of the RIF (R. Wischnewski), CIROS has not previously been used for the modelling and simulation of a transportation system with active (self-driving) carriers on passive tracks. The original implemented use of the Transport Simulation was for the modelling and simulation of longitudinal transfer systems with friction generated motion of passive carriers on active tracks such as Bosch Rexroth TS2plus with permanent belt drive (Wischnewski & Rossmann, 2010). A publication by Kan Li describes a case study at Tampere University, realising the simulation of a pallet-based production line with the pallet conveyor Dynamic Assembly System (DAS) by FlexLink Automation (Li, 2011a). Thus, when starting the modelling of the CTS transportation system, it became necessary to add basic elements needed for the CTS, which has been carried out at the author's request by a developer of the RIF.

The following hypothesis arises out of these facts and the abovementioned features:

> *It is possible to model the Bosch CTS transportation system with its self-driving transport cars, on passive tracks, using CIROS Studio and to conduct a VC incorporating the STEP7 program and the WinCC Flexible HMI application.*

The testing of this hypothesis is presented as part of the chapters six and seven.

5.2.3 Environment for the VC of the RFAS

The indispensable specification of an appropriate VC environment occurs in procedural step 2 (Project Planning) of the proposed methodology for a VC simulation study (cf. Fig 3-1). After careful consideration based on the functionality of CIROS, its extension modules and the existing hardware of the RFAS, the CIROS environment with its sub-systems illustrated in figure 5-5, has been projected to be functionally appropriate for the VC of the entire system. This applicability to the entire system is not a matter of course (cf. subsection 5-1), but due to the ongoing research on VC, this has been defined as a requirement. However, a VC of the robot cells as well as the VC of the entire system will be postponed to future work.

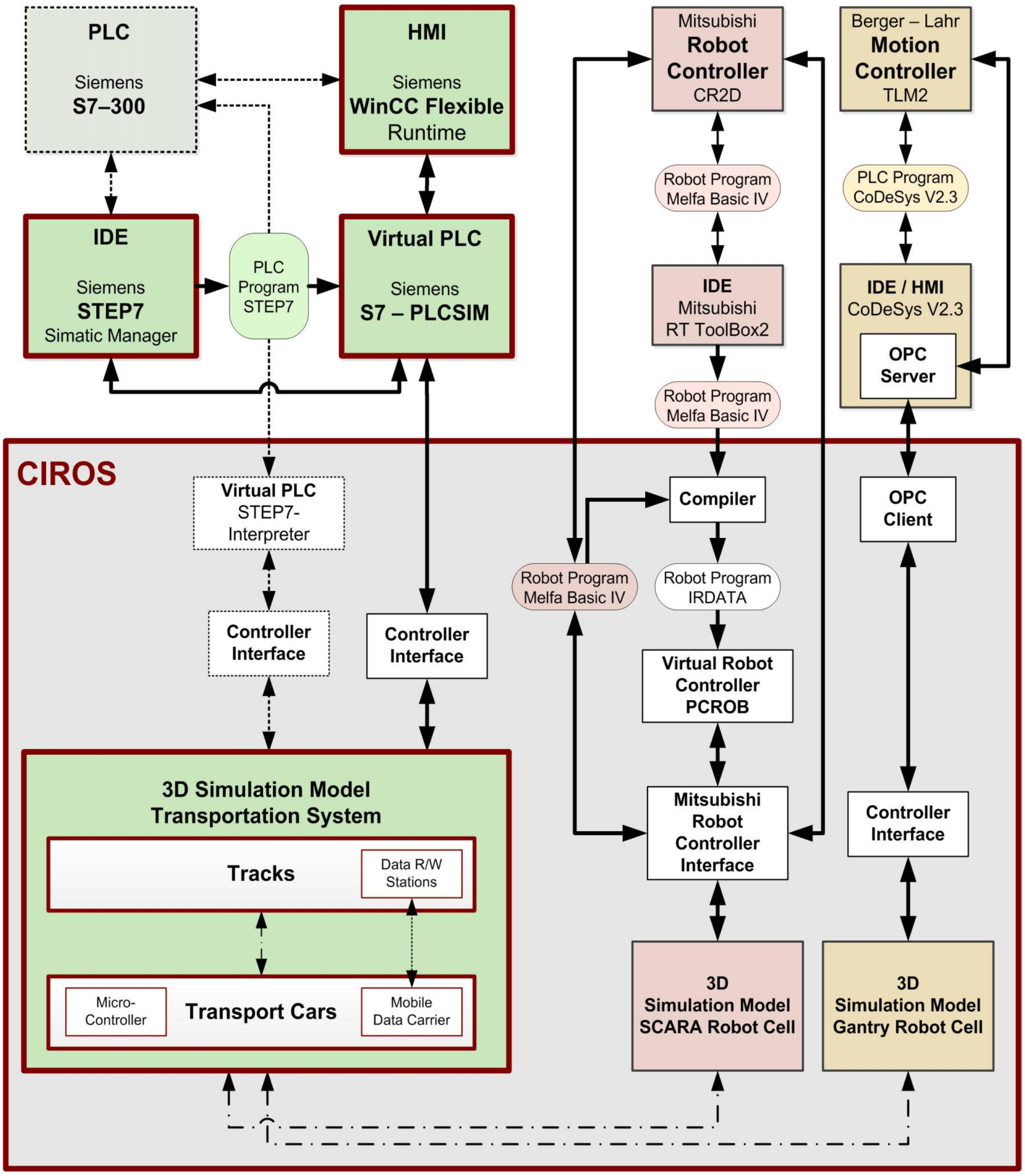


Fig. 5-5: Projected environment for VC of the RFAS based on CIROS

The green boxes with red borders (HMI WinCC Flexible Runtime, Virtual PLC S7-PLCSIM, IDE STEP7 and CIROS with the 3D simulation model of the transportation system) indicate the parts of the complete environment that are relevant for the VC project in this thesis. This configuration with SIL simulation utilising S7-PLCSIM as virtual controller executing the STEP7 program for the control of the 3D simulation model of the transportation system has been selected for the VC in this thesis. It allows the advantageous coupling of the WinCC Flexible Runtime with PLCSIM and consequently operator inputs via the original control displays are possible. Neither a real PLC nor the real touch panel are necessary to test the original PLC programs and the control displays at the same time. The functionality of the internal virtual PLC of CIROS will be tested during implementation, although it does not allow the coupling with WinCC Flexible.

In principle, a HIL simulation using the real S7-PLC would be possible too. In this case an OPC server for S7 (e.g. from SIMATIC Net) would be necessary additionally, which allows coupling with the CIROS OPC client.

5.3 Simulation model building for VC

An essential requirement for a VC project that allows for the V&V of control programs in conjunction with planned mechanical setup is the 3D simulation model of the manufacturing system, which is generally not available at the beginning of VC. This mechatronic plant model has to be generated for the selected plant simulation system in a specific simulation model building procedure.

The building and validation of simulation models can be a particularly severe problem if the real-world manufacturing system has not yet been built (cf. subsection 2.3.4). In this study, the hardware components and sub-systems described in chapter 4 are available for testing and validation of simulation models, which is not the rule. It is usually the case that the VC needs to be conducted before assembling and commissioning the new manufacturing system; otherwise, it has no direct benefit. If no opportunity offers itself to validate the models using a similar system prior to an upcoming VC, errors that appear are not explicitly attributable to a new simulation model or any untested controller programs.

Manufacturing systems such as the RFAS are generally composed of many off-the-shelf components and some purpose-built components. Basically, the transportation system is assembled from off-the-shelf components (Fig. 5-6) by Bosch (cf. subsection 4.1.2). It is not uncommon to have a high percentage of standard parts in such a system. Only the length of straight-line tracks and the additional mounting plates on transport cars are user specific. The SCARA robot, the single axes of the gantry robot, all grippers (with exception of some gripper jaws) and the automatic gripper changing systems consisting of gripper adapter for the robot and for the grippers are standard components, but other peripheral components in the robot cells e.g. the parts provisioning are purpose-built.

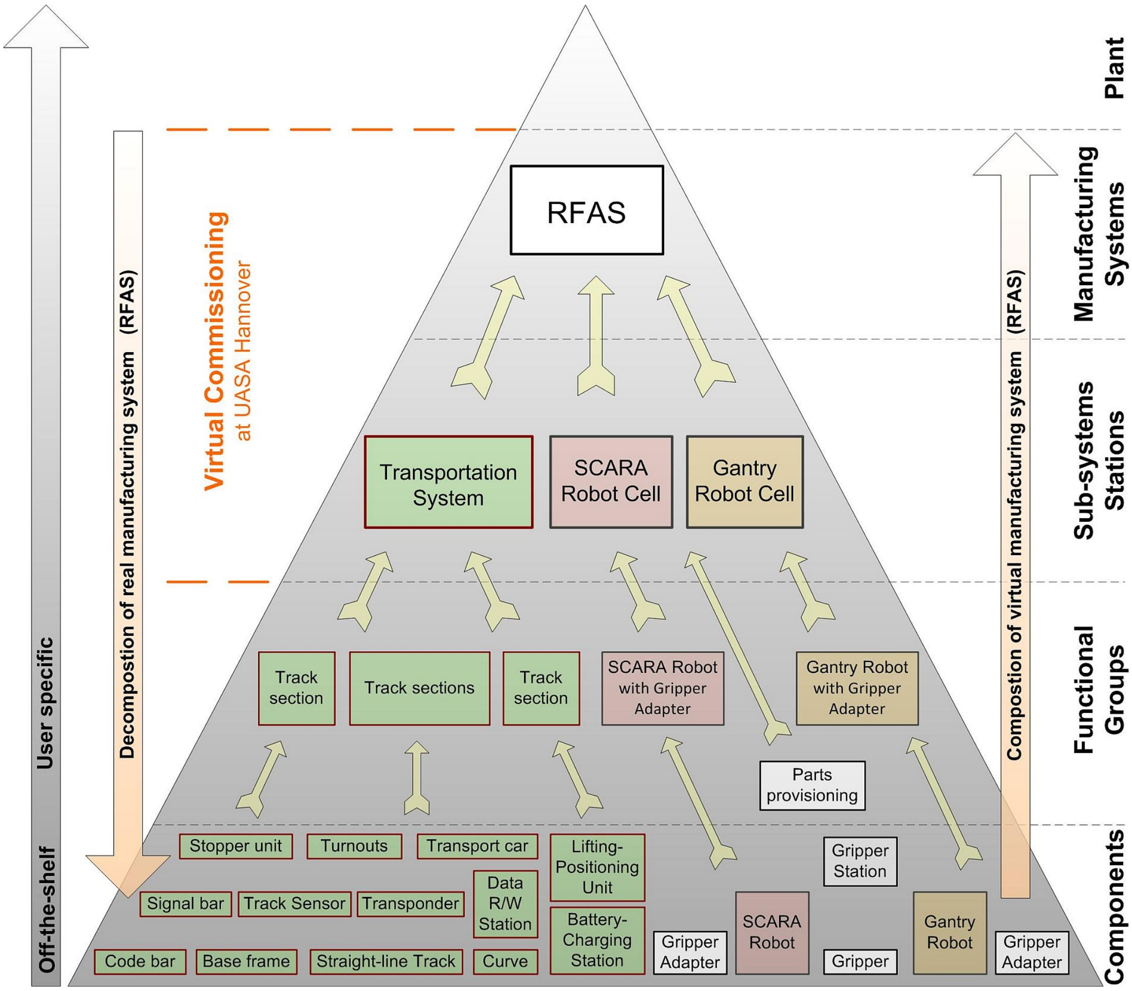


Fig. 5-6: Hierarchical levels for simulation model building of the RFAS

The off-the shelf components (Fig. 5-6) are characterised by a high re-usability, the more user specific and also more complex the models get, the less re-usable they are. A modular, component-based approach has been applied to the simulation model building in this thesis by transferring the components from decomposition of the real transportation system to equivalent component models for simulation.

In principle, it would be possible to build the simulation model of the transportation system or even the entire RFAS without the proposed previous decomposition. The 3D CAD data of the whole system could be transferred to CIROS in its entirety and the simulation model building could be done based on the imported system as a whole. Such an approach would have several disadvantages or rather one could not benefit from the advantages a component-based approach provides. Disadvantages of a non-modular approach are for example:

- The simulation model building and VC of a complete manufacturing system such as the RFAS (transportation system and divers robot cells) started from scratch would result in too much effort and excessive complexity

- The modeller would be forced to always handle the complete system during work inside the modelling editor
- The modelling would have to be repeated for identical components that are included numerous times in the system (not unusual for e.g. transportation systems). A modeller would rapidly think about easing his workload by reuse in the form of “copy and paste” or component libraries
- It would be necessary to verify and validate the complete simulation model as a whole

Advantages of the component-based, modular simulation model building approach are:

- Reusability: If the simulation tool provides a library concept (as CIROS does) the component models can be easily reused for future application to new systems
- Flexibility: A manufacturing system model composed of component models can be easily redesigned (e.g. changed layout of a transportation system)
- Testability: The component models can pass through V&V separately, and the V&V of complete system after composition from component models is much less complex based on already tested component models

Last but not least, a component-based approach supports the proposed novel model collection methodology (cf. subsection 5.4.2).

Besides, it is possible to increase the benefit from a modular approach if each component of certain complexity is provided with a dedicated control module such as function blocks in PLC programs (cf. subsection 4.6.1), which can be tested separately as well, possibly with the related simulation component model if applicable. Possible drawbacks of the component-based approach compared to a procedure without decomposition are discussed in chapter 8.

The standard components are often provided by the manufacturers with appendent 3D CAD data today, but applicable CAD data for the mechatronic components of the transportation system used at the UASA Hannover had not been available and so considerable effort arose from the necessary building of 3D CAD models.

Thus, the starting point for the simulation model building procedure is the 3D CAD data, in most cases available from the manufacturers of implemented off-the-shelf components, or in the case of purpose built items generated during mechanical engineering by an in-house department of the operating company or by third-party manufacturing system builder.

5.3.1 High-level Plant modelling and Low-level Component modelling

The simulation model building procedure for stand-alone simulation tools such as CIROS (which means not integrated in Digital Factory suites of tools, and not requiring a co-simulation with a second simulation tool) varies in detail from tool to tool. Especially the modelling of components' behaviours differs between tools, but this procedure has some general principles in common, e.g. the typically necessary CAD preparation and the general simulation model building procedure for 3D mechatronic plant models. As presented in Fig. 5-7, this model building can usefully be divided into the two different modelling tasks "High-level Plant Modelling" and "Low-level Component Modelling". This principal of modelling task splitting, proposed in (Hoffmann et al., 2012), has been taken up and supported by other work (Dzinic & Yao, 2013, Cruz et al., 2014a).

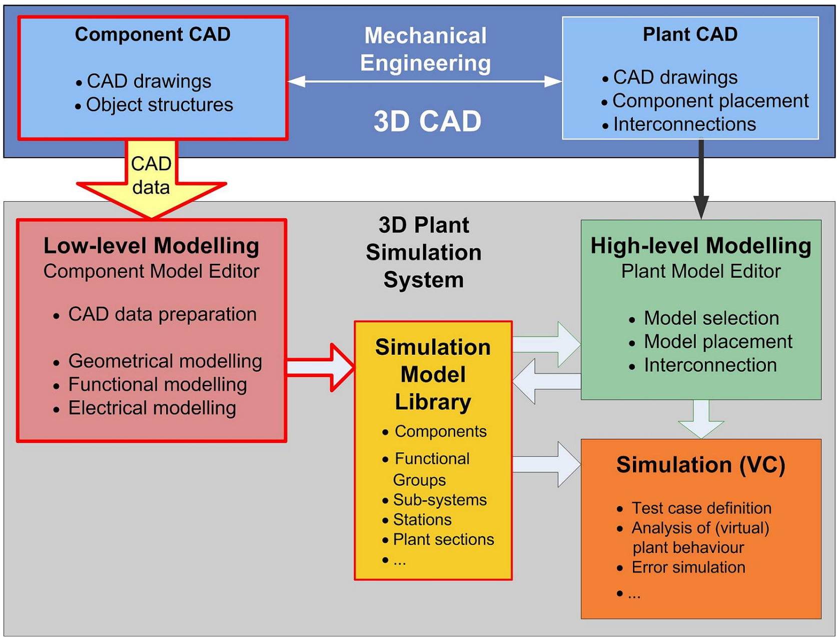


Fig. 5-7: Simulation model building procedure for VC

As highlighted (red borders) in figure 5-7 the Low-level Component Modelling serves as a procedure to fill the simulation model library with the required mechatronic component models. This is the critical task in current approaches to VC. The 3D CAD data delivered from the manufacturers of components and sub-systems or generated during plant design are not in general directly suitable for use in a VC simulation because most CAD models are geometrically too complex and/or unsuitable structured. This is why, a thorough model analysis and pre-processing of the CAD model data is usually necessary to reduce calculation loading and to allow a proper simulation model building based on these CAD data. The goals are a reduced number of details (geometrical elements) resulting in a reduced number of facets in the simulation model and good handling of 3D models in the model editors of simulation tools.

5.3.1.1 High-level Plant Modelling

Many 3D simulation tools allow the aggregation and simulation of virtual manufacturing systems based on their internal simulation model library providing several predefined mechatronic component models. These ready to use library models already contain the functional interaction of mechanical behaviours with actuators and sensors, and are sometimes called "Smart Components" (ABB RobotStudio), "Smart" catalogue elements (Emulate3D) or "Smart Devices" (Delmia Automation).

If such mechatronic component models, needed for the VC of a specific plant, are already contained in the simulation model library it is possible to compose the mechatronic plant model for VC from these component models using the 3D editor of the simulation tool. This task can extend to more than one level in figure 5-6, namely from functional group level to the manufacturing system level and rarely to the plant level. It is possible to compose a functional group (e.g. a track section of the transportation system) from library components (e.g. straight-line track, stopper-unit) and store this functional group in the library afterwards. Subsequently a sub-system such as the transportation system or other stations (e.g. robot cells) can be composed from such functional groups (and also be stored in the library if applicable), followed by the aggregation of the manufacturing system from sub-systems and stations. This is the top level at the UASA Hannover, but it is principally possible to extend this task to plant level by combining several manufacturing systems.

By the means of this High-level Plant Modelling, it is relatively easy to set up the mechatronic plant model and to conduct a VC, but some additional effort arises from the necessary configuration of the interconnections (mechanical, electrical I/O). Some state-of-the-art simulation tools such as CIROS support the exact placement of components by a "snapping" functionality for e.g. easily attaching the track components of a transportation system.

However, nowadays the geometrical information for the exact placement of the components, functional groups and sub-systems (e.g. robot position relative to transportation system) and for their connections must be transferred from the CAD drawings provided by mechanical engineering for the plant (Fig. 5-7) to allow for an aggregation and interconnection of component models to a plant model. The lists of I/O signals for the mapping between actuators and sensors of mechatronic plant model and (virtual) controller I/O's has to be transferred from the controller programs.

5.3.1.2 Low-level Component Modelling

Generally, appropriate simulation models of all required components or sub-systems are not available in the simulation model library when starting with VC. Thus, it is necessary to create new mechatronic component models from available 3D CAD data in a multistage procedure. This task is relevant for the lowest level in figure 5-6, the component level.

It is necessary to carry out geometrical, functional and electrical modelling to create a structured mechatronic model, thus Low-level Component Modelling comprises the whole functional chain (cf. subsection 5.3.4) and is a non-trivial task requiring considerable modelling expertise. As a consequence of this it is difficult to set up a VC if needed mechatronic component models are not yet available and remarkable efforts become necessary then for this Low-level Component Modelling. These efforts are far greater than for High-level Plant Modelling and are meant if “high modelling effort” is complained in literature (cf. chapter 2), and are a main reason for preventing especially SMEs from conducting a VC.

5.3.2 3D-CAD Data Transfer – the kinematics problem

The transfer of 3D CAD models from CAD tools to simulation tools has other requirements than the data exchange between different CAD/CAM tools used in manufacturing engineering. For VC are e.g. detail data like surface finish specification or tolerances not necessary, but material data in contrast may make sense, e.g. for the simulation of sensors with different measurement principles as partly provided in simulation tools such as CIROS.

A big problem exists to date at the interface between the different CAD tools used in mechanical engineering (from either external component manufacturers or in-house CAD) and simulation tools for VC. Sometimes 3D models for mechanical engineering will already be equipped with kinematic aspects like specification of axes and joints (motion constraints, degrees of freedom). Even if such functionally structured CAD models are available, the standard data interfaces currently available allow in general only partial transfer (without kinematics) of this structure information to simulation tools such as CIROS (q.v. table 5.1).

Even the transfer of all other information included in the CAD models such as part naming, material properties or colour is not assured. Exported STEP (ISO, 1994) models with standard option (class VI, multi volumes) from Solid Edge could be imported in CIROS with preserved scale, colours and part names, but it has not been possible by any tested option to transfer material properties from Solid Edge to CIROS. The additional selection of STEP application protocol AP214 (ISO, 2010) in the export filter of Solid Edge does not bring an advantage for the CIROS import. The tests of the transfer from Creo Parametric to CIROS resulted in preserved scale and part names, but colours and again material properties could not be transferred with any tested option.

In order to transfer kinematics additionally to geometry it is in principle possible to use API functions for access to internal data of the CAD tool. Such a solution was demonstrated by (Neugebauer & Schob, 2011). The disadvantage of such approach is the necessary programming and software maintenance (e.g. necessary for every new version of one involved tool) of each combination of CAD tool and simulation tool. It is proposed that it is more promising to develop suitable standardised CAD exchange formats further, than to develop specific kinematics exchange support for every combination of used CAD and simulation tools. There are many established exchange formats such as STEP (**ST**andard for the **E**xchange of **P**roduct model data) and JT (**J**upiter **T**essellation) with special advantages available (Beckers et al., 2010, Fröhlich, 2013, Friedewald et al., 2011, Kunze, 2012), but none are established for exchange of kinematics with simulation tools for VC.

In principle, the widely used STEP format is suitable for the exchange of geometry and kinematics, because STEP has supported kinematics since an ESPPRIT research project in 1996 (Haenisch et al., 1996), standardised as “Part 105, Integrated application resources: Kinematics”. The AP214 allows open and closed kinematic chains in different structures and supports the modelling of forward and inverse kinematics, which is principally useful for simulation (Kjellberg et al., 2009), but to date there is no industrial implementation in a tool. Generally, CAD tools neither export STEP with kinematics nor are simulation tools able to import STEP with kinematics today. First approaches for CAD to support STEP AP214 with kinematics based on Siemens NX are presented in (Li et al., 2011, Hedlind et al., 2011, Li, 2011b, Li et al., 2015).

A comparatively new approach is the development of STEP AP242 XML (ISO, 2014). This aspiring standard aims at the merging of the older application protocols AP203 (ISO, 2011) and AP214 and their substitution by a new common standard for automotive and aerospace industries (AP242, 2016). Compared to AP203/AP214 it shall provide STEP with new or considerably enhanced features such as 3D tessellated geometry, 3D parametric and construction history, 3D assembly constraints and, importantly for simulation, improved kinematics (ProSTEP-iViP, 2016). The website of the ProSTEP iViP association remarks that currently no products implementing this standard are known, but at least one conversion tool, COM/FOX developed by T-Systems for Daimler, is available for data transfer between CAD tools CATIA and NX using JT and AP242 XML for kinematics (automotiveIT, 2015). Another format principally suitable for the transfer of geometry and kinematics to simulation tools is COLLADA utilised by AutomationML, but COLLADA 1.4.1 (without kinematics) is not widely supported by CAD tools or simulation tools to date, and COLLADA 1.5 with kinematics even less so. Due to the likewise XML based concept of STEP AP242 it would be possible to reference STEP AP242 in a similar manner to the practice with COLLADA files in AutomationML as proposed in (Schlögl, 2012). This possible exchange of kinematic data using XML with AP242 might be an interesting approach for simulation and VC in the future.

As a result, the transfer of 3D CAD models with kinematics from CAD to simulation tools is not possible in most cases today. In (Hollander & Sappei, 2011) the authors for example complain *"No opportunities to convert more than the graphical representation and measurements were found when transferring the CAD files from Pro/ENGINEER to any of the investigated simulation tools"*. They investigated the simulation tools Delmia V6, Visual Components 3D-Create, Siemens Process Simulate and FlexSim, several CAD conversion tools, and tried native CAD formats as well as STEP and JT. The impossible transfer of kinematics between Pro/Engineer and Delmia Automation has already been a complaint in (Davidson & Sennö, 2005), but the authors stated one of the few exceptions; if a model is created in Dassault's CAD tool CATIA the transfer with kinematics to Delmia Automation has been possible (both are part of same digital factory suite of tools). From the automotive industry (Daimler), the exchange of kinematics between a robot simulation tool (not specified) and INVISION, a simulation tool for VC, via AutomationML-COLLADA has been reported to be possible in (Grimm, 2012).

Generally, it is necessary to (re-)model the kinematics of mechatronic components during simulation model building today, best done if CAD models are already well structured. A systematisation of this procedure is given below.

5.3.3 Proposal for a systematic simulation model building procedure for VC

Nowadays many component manufacturers provide 3D CAD data of the components, but as demonstrated by the examples above, they are not necessarily usable for simulation. Thus, for all components not available in the simulation model library of the simulation tool, a CAD preparation and a subsequent Low-level Component Modelling procedure must be carried out by a modelling and simulation expert. The 3D CAD models of purpose-built components and sub-systems generated during in-house mechanical engineering will probably show likewise problems regarding complexity and structure, if a re-use of CAD data for simulation has not been considered by mechanical engineers. In this study, the Low-level Component Modelling procedure has been critically reviewed, because especially this task complicates the simulation model building for VC considerably and requires specific modelling expertise of the entire simulation model building process. Therefore, the Low-level Component Modelling procedure needs systematisation, especially the formulation of requirements regarding function-oriented structuring of CAD models by design engineers is crucial.

After analysis of necessary processes the following systematic workflow (Fig. 5-8) for simulation model building of mechatronic plant models based on component CAD models has been proposed (Hoffmann et al., 2012). The mechatronic component model building (see Fig. 5-9), including CAD preparation and Low-level Component Modelling, has to be repeated for every required component missing from the library, whereas the High-level Plant Modelling (Fig. 5-18) is a one-time task.

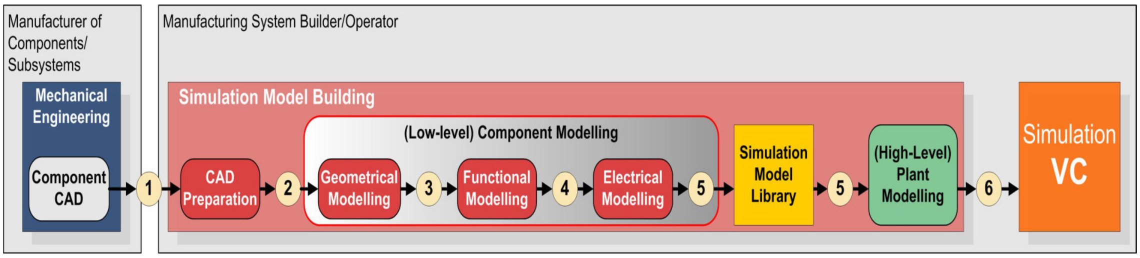

Manufacturer of Components/ Subsystems
Mechanical Engineering
Component CAD
Manufacturing System Builder/Operator
Simulation Model Building
(Low-level) Component Modelling
CAD Preparation
Geometrical Modelling
Functional Modelling
Electrical Modelling
Simulation Model Library
(High-Level) Plant Modelling
Simulation VC
1
2
3
4
5
5
6

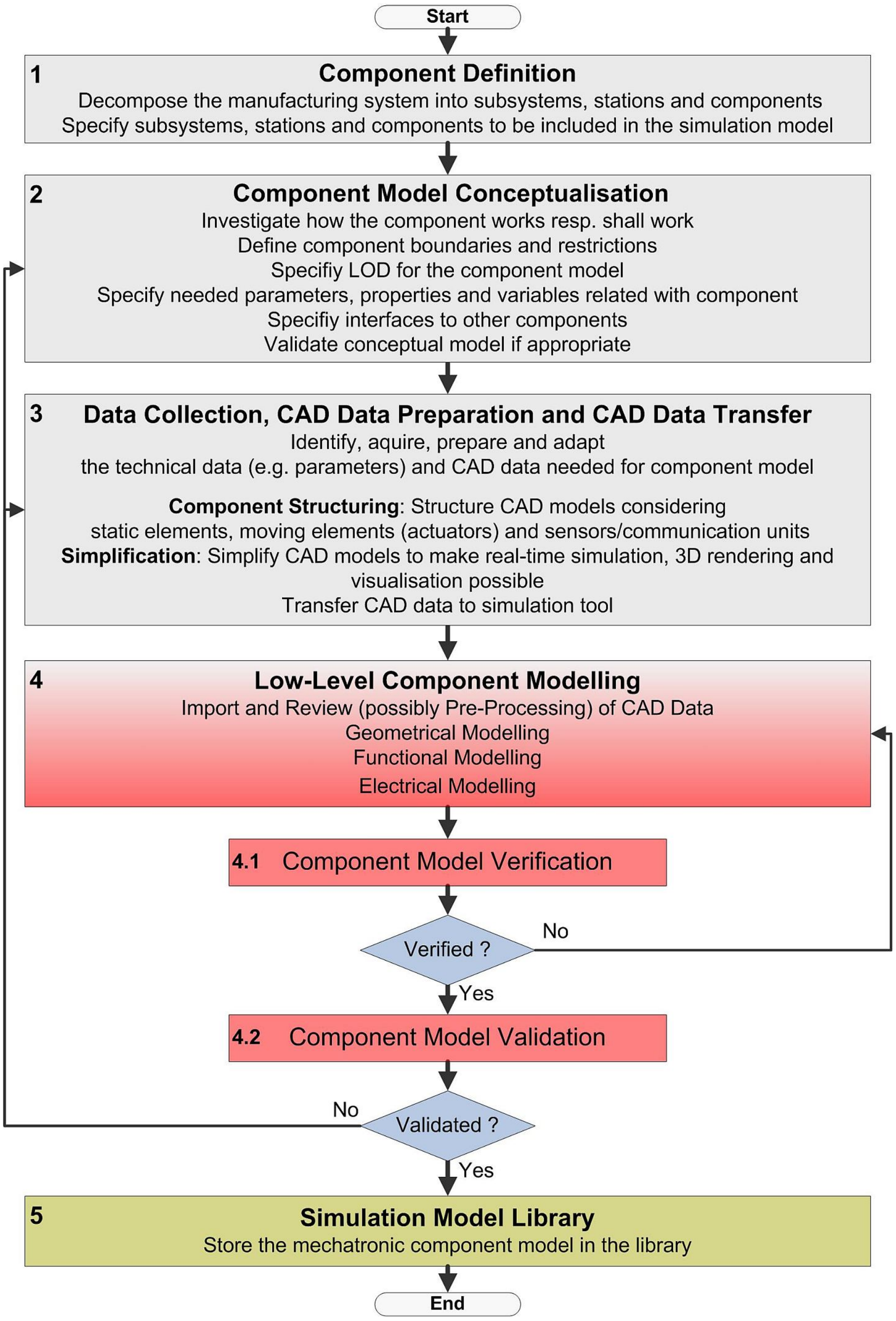
Fig. 5-8: Workflow from component CAD model to mechatronic plant model

First, an overview of simulation model building workflow, based on the type of generated model data (Fig. 5-8, ①-⑥) is given. The CAD preparation and the Low-level Component Modelling with its three sub-tasks are described below in detail. In CIROS the workflow for the building of mechatronic component models for simulation starts with 3D CAD models intended for mechanical engineering (①) provided by the component manufacturer (or in-house mechanical engineering). The company (manufacturing system builder/operating company) that conducts the simulation model building pre-processes these CAD models regarding complexity and model structure during CAD preparation (see below). This thesis proposes to conduct the CAD preparation already in the used 3D CAD tool. The outcomes are simplified and adequate structured CAD models for VC (②) in native CAD formats or exchange formats (e.g. STEP). If the 3D simulation tool provides the necessary features for this purpose, it can be also conducted there. Thus, the geometrical modelling conducted in the simulation tool is interrelated to the CAD preparation; an adequate CAD preparation reduces the geometrical modelling effort. Due to effects of CAD tool export filters and simulation tool import filters (e.g. because of parameter choice at export and import) a geometrical modelling can become necessary, despite of an appropriate CAD model. The outcomes of this task are suitable structured geometrical component models in the internal format of the simulation tool (③). Having created appropriately structured geometrical models, these must be equipped with functionality (Functional modelling) respectively component behaviour, involving kinematics, actuators and sensors. This task is quite varying in detail in different simulation tools, the detailed description below bases on CIROS. The results are functional component models (④) with kinematics, actuators and sensors if applicable. These functional models are supplemented with electrical input and outputs in the electrical modelling completing the Low-level Component Modelling. The resulting mechatronic component models (⑤) can be stored in the simulation model library. The mechatronic plant model (⑥) for VC is composed of this mechatronic component models during the High-level Plant Modelling.

This specified workflow, especially the included CAD preparation (see below) is generally applicable, not only to CIROS, but to other comparable 3D simulation tools too. Besides its application to the simulation model building for the RFAS in this study, it has also been applied in a case study for the Swedish company Teamster AB which has been carried out at Chalmers Production Systems Laboratory (Dzinic & Yao, 2013). The authors adapted the workflow proposed in (Hoffmann et al., 2012) to its use with the 3D simulation tool Xcelgo Experior and SketchUp for CAD preparation.

5.3.4 Systematic model building of mechatronic components for simulation - Low-level Component Building

Based on the steps 3-5 of the proposed simulation study methodology for VC (Fig. 3-1) the following proposal for systematic mechatronic component model building (Fig. 5-9) has been developed.



Start
1 Component Definition
Decompose the manufacturing system into subsystems, stations and components
Specify subsystems, stations and components to be included in the simulation model
2 Component Model Conceptualisation
Investigate how the component works resp. shall work
Define component boundaries and restrictions
Specifiy LOD for the component model
Specify needed parameters, properties and variables related with component
Specifiy interfaces to other components
Validate conceptual model if appropriate
3 Data Collection, CAD Data Preparation and CAD Data Transfer
Identify, aquire, prepare and adapt
the technical data (e.g. parameters) and CAD data needed for component model
Component Structuring: Structure CAD models considering
static elements, moving elements (actuators) and sensors/communication units
Simplification: Simplify CAD models to make real-time simulation, 3D rendering and visualisation possible
Transfer CAD data to simulation tool
4 Low-Level Component Modelling
Import and Review (possibly Pre-Processing) of CAD Data
Geometrical Modelling
Functional Modelling
Electrical Modelling
4.1 Component Model Verification
Verified ?
No
Yes
4.2 Component Model Validation
Validated ?
No
Yes
5 Simulation Model Library
Store the mechatronic component model in the library
End

Fig. 5-9: Procedure for systematic model building of mechatronic simulation components

According to (Wischnewski & Freund, 2004), this modelling procedure for mechatronic simulation components is intended for "*technically experienced persons with detailed knowledge of mechanical, electrical, pneumatic, and geometric data of single components*". Consequently, this task should preferably be done by modelling experts from the component manufacturer who generally do not exist today, because the manufacturers have the necessary product knowledge but often lack the modelling expertise. In fact, modelling has to be conducted by potential simulation users themselves or service providers have to be commissioned to do this work.

The proposal for this systematic procedure has been developed considering CIROS, but it is applicable and useful for other comparable tools too. The steps are elaborated below.

5.3.4.1 Component Definition

Component definition (Step 1) is attended by the decomposition of the manufacturing system into subsystems, stations and components and analysis as to which of them are relevant for the VC. Every real component supposed to be relevant for VC must be included in the plant simulation model and so it must have a correspondent component model. If such component models are already available in the library of the simulation tool, they can be directly used for the High-level Plant Modelling, for all other components, a repeated execution of steps 2-5 become necessary.

5.3.4.2 Component Model Conceptualisation

During conceptualisation (Step 2), recommended by several researchers (cf. subsection 2.3.3), the component functionality has to be explored as part of the entire system. Investigation as to how the component works or should work is related to the conceptualisation of the entire manufacturing system (Fig. 5-18). As a result, one gets the definition of component boundaries and restrictions and the specification of interfaces between components. Besides determination of component detailing, the analyst must specify the required properties and parameters.

The depicted overall concept for the simulation models of tracks and transport cars in figure 5-10 gives an example for the proposed component model conceptualisation (Fig. 5-9, Step 2) and the specification of the functional relationship between components during model conceptualisation, in procedural step 1 of the proposed High-level Plant Modelling procedure (Fig. 5-18).

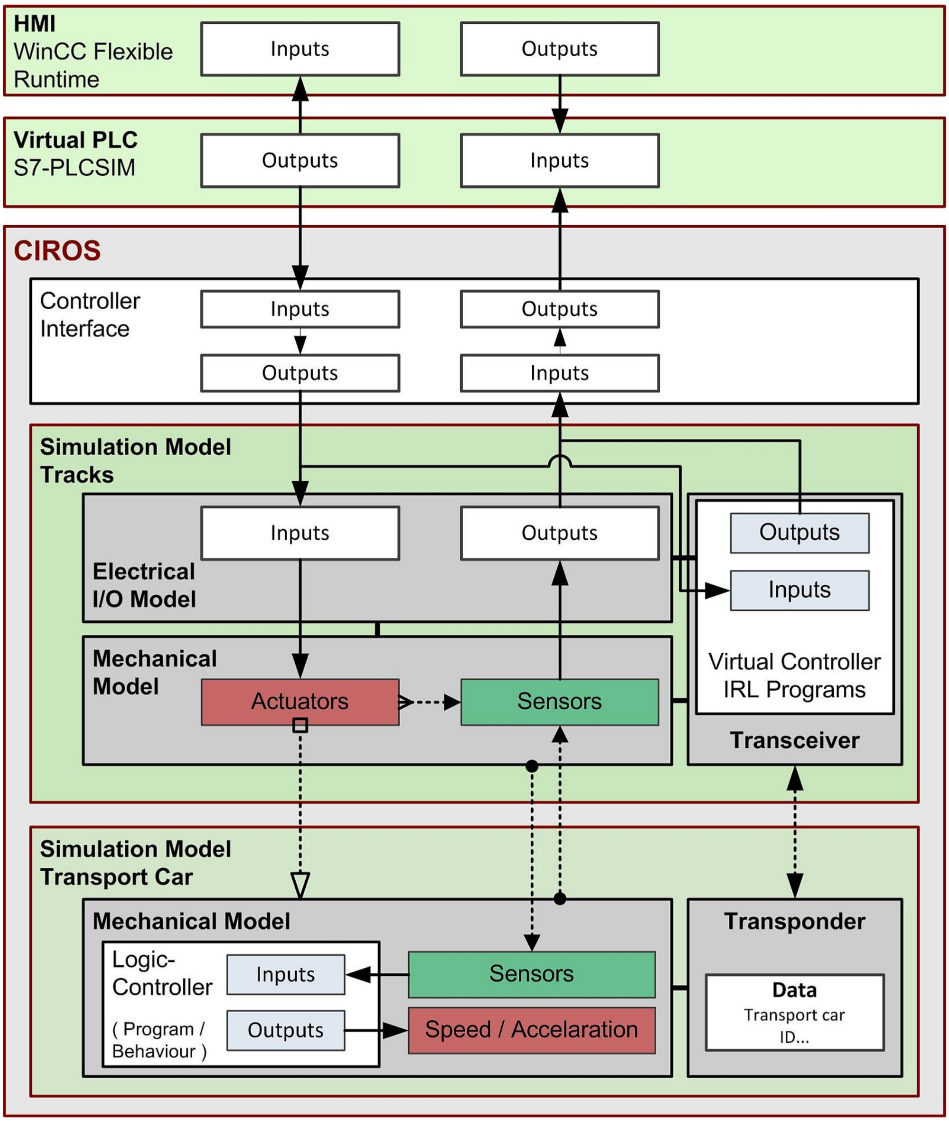


Fig. 5-10: Model conceptualisation of transportation system and functional chains in VC

The transport car has a complex internal logics behaviour (cf. chapter 4). Because of this complexity, an individual conceptual model for the transport car makes sense. Hence, such a model with particular consideration of logics/behaviour, has been developed (cf. subsection 6.1.1.4).

5.3.4.3 Data Collection

In step 3, the modeller should identify, acquire, prepare and adapt technical data and CAD data needed for component models. The technical data include component parameters (e.g. speed of an actuator or measurement range of a sensor) and for complex components, behavioural descriptions if available.

3D CAD data are the basis for the building of the simulation models, and CAD preparation is a crucial step for the design of mechatronic component models and thus important for VC.

Web portals such as e.g. CADENAS PARTsolutions (Cadenas, 2014) or Traceparts (Traceparts, 2014) and the web pages of manufacturers provide CAD data of millions of components for download. These models are provided in several native CAD tool formats or exchange formats like e.g. STEP. The review of such 3D CAD drawings delivered from manufacturers of components shows two problems regarding the use of these models for simulation purpose: 3D CAD models from manufacturers are often geometrically too complex and/or unsuitable structured.

For example, the provided CAD data of aluminium profiles (Fig. 5-11), used for many constructions within manufacturing systems such as the RFAS at the UASA Hannover, must definitely not be used directly for simulation in this detailed from, either a simplification or a redesign is necessary if no simplified CAD data is available for download.

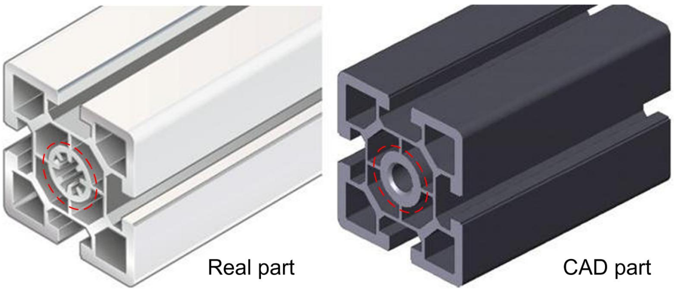


Fig. 5-11: Aluminium profile – Real and CAD

Figure 5-11 shows an exemplar aluminium profile and its CAD data provided by the manufacturer, and downloaded in detailed STEP format via Traceparts (Bosch Rexroth STRUT PROFILE 60X60). Even if this CAD model is simplified (Fig. 5-11 red circle), it is still overly complex.

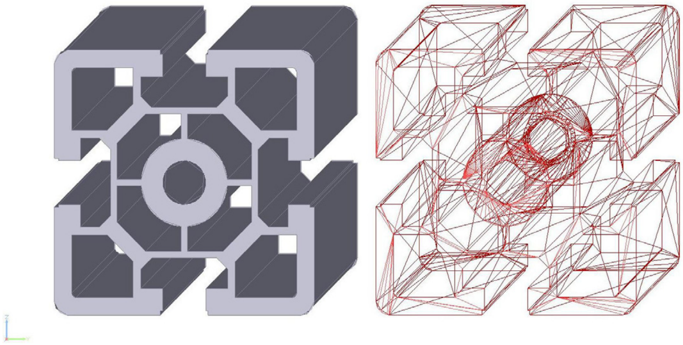


Fig. 5-12: CAD data aluminium profile with facets

Complex geometric structures such as those in this aluminium profile (Fig. 5-12) would greatly increase the number of facets and extend the calculation time for simulation, because many of such profiles are typically used. This example is not as complex as the aluminium profile used for the tracks of the transportation system (Fig. 4-16). This aluminium profile illustrates the necessity to avoid such complex model structures in general.

The complexity of CAD data is not the sole problem in simulation; the inner structure of a CAD models has often to be changed as well. The following simple examples for components of the transportation system TS2Plus had been downloaded from (Bosch-Rexroth, 2014) as STEP file without providing assembly merging as an option. The Bosch Rexroth's download portal also provides e.g. IGES, CATIA, DXF and VRML but not the native CAD formats for the used CAD tools Solid Edge and Creo Parametric. Figures 5-13 to 5-15 show the assemblies in Solid Edge. An example for an unsuitable model structure is the following stopper unit, similar to the stopper at the RFAS, with moving parts not separated in CAD model (Fig. 5-13).

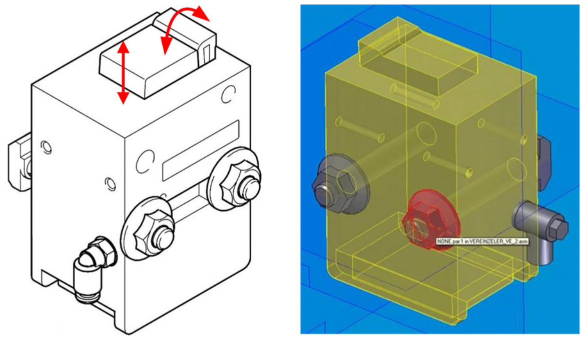


Fig. 5-13: Current CAD data for stopper unit VE2 from Bosch Rexroth transfer system

Another example for unsuitable structured CAD models is the following lifting-positioning unit HP2 that is available as detailed version (Fig. 5-14 left) and as simple version (Fig. 5-14 right). The tested STEP models of this HP2, downloaded as AP214 and AP203, show only different grey colours but no significant differences.

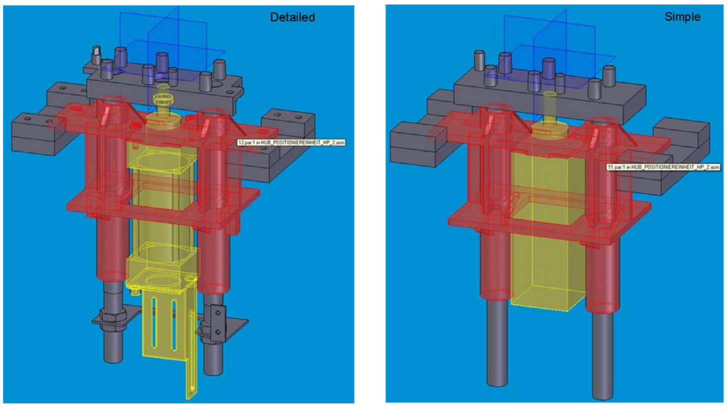


Fig. 5-14: Current CAD data for lift positioning unit HP2 from Bosch Rexroth transfer system

The simple version could be suitable for simulation, but therefore, the separation of cylinder and piston would be necessary, which is not the case in the CAD model provided (yellow part). It would be useful for simulation model building, if e.g. the piston and the adapter plate would be one part. Without separation of the moving parts in CAD, the use off such components as moving actuator in a simulation model is not possible. Additional tests for this CAD component, downloaded in "Pro/Engineer neutral" format via Traceparts (Traceparts, 2014), within Creo Parametric does not show better results with regard to simulation use.

The 3D CAD model of the FESTO pneumatic cylinder in figure 5-15 was downloaded as STEP AP214 assembly via Traceparts (Traceparts, 2014).

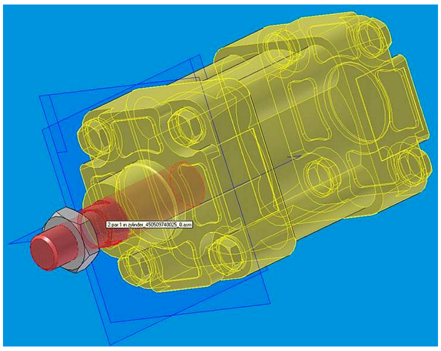


Fig. 5-15: Current CAD data for FESTO pneumatic cylinder

In contrast to cylinder in figure 5-14, this model has a moveable piston, but it has been available as CAD version with detailed housing only, which would not be suitable for the use as part of mechatronic simulation models without simplification.

5.3.4.4 Requirements for CAD Data Preparation

The complexity and model structure of CAD data for mechanical engineering is mostly not directly suitable for its application in simulation; this often inhibits the unimpeded use of this 3D CAD data for VC today. As a consequence, the preparation of 3D CAD data for simulation has two requirements.

1. The first requirement is related to the available computational power. The CAD data complexity (number of components and facets) must be reduced by simplification to make real-time simulation, 3D rendering and visualisation possible, especially if the time response of simulation depends on the 3D graphics calculation. According to (Bergert et al., 2009) this is the case in some simulation tools like e.g. Delmia Automation and Siemens Process Simulate Commissioning. Even if the simulation tool is able to calculate independently the 3D visualisation, the visualisation should not differ that

much. Deviation would hinder the human visual analysis of the operations in the virtual manufacturing system during VC. Some simulation tools such as e.g. CIROS provide such a possibility (cf. subsection 5.1.1, Fig. 5-4).

2. The second requirement is related to a principal problem that cannot be solved by means of more computational power. The model structure must consider actuating elements and sensors. The dependencies of objects moving together/separately or being stationary, and which geometrical objects are sensors, are all relevant and important. This often necessitates a component structuring procedure (see below) for CAD models.

5.3.4.5 CAD Data Preparation - Simplification

The simplification of CAD data can be divided into three related operational tasks: removing parts (object filtering), removing features and mesh simplification (Hoffmann et al., 2012).

1. Object Filtering
   Many objects are not necessary for simulation and the removal of irrelevant geometrical objects is recommended. Typical objects to be removed are:
   - Hidden invisible parts (e.g. insides of objects)
   - Small objects (e.g. small thread bolts, screws and nuts) or other selected parts

Instead of a complete removal, it is possible to replace the numerous standard or norm parts (e.g. screws and nuts) used in CAD during mechanical engineering by simplified geometries, which is supported by automatic functions in CAD tools (Strahilov et al., 2012). The outcome of using many detailed norm parts (e.g. chamfered (internal) hexagon screw heads) would be a large number of facets, thus a replacement by simple geometries makes sense. In contrast to a complete removal, the number of parts (which is also relevant to the required computational power) is not reduced, but dependent on the dimension of the parts it might be preferred.

2. Feature Suppression
   It is recommended to reduce the detailing (of e.g. roundings) or to completely remove geometrical features that are irrelevant for simulation. CAD tools allow a parameterisation for these tasks, e.g. only holes smaller than a defined diameter will be removed. Typical items to be reduced or removed are:
   - Holes, bosses, pockets, breakthroughs (not round)
   - Fillets, chamfers, roundings
   - Ribs, steps, slots
   - Labels and logos

The feature suppression and object filtering will automatically result in a reduced number of facets.

3. Tessellation and mesh simplification

The original CAD model is tessellated into a mesh of polygons, often done by triangulation to build a triangular mesh, e.g. for approximation of freeform surfaces. Generally, it is essential to reduce the quantity of polygons for simulation purposes. The research in (Kuhlenkötter et al., 2010) shows the influence of quantity of facets and type of mesh (polygonal/triangular) on calculation time using ABB RobotStudio as example. Unsurprisingly, the calculation time for collision detection ascends with quantity of facets, but the conducted experiments offer big differences between polygonal meshes and triangular meshes. Simulation tools for VC make use of collision detection between geometrical objects, and triangular meshes show the advantage of significant shorter calculation time. These findings allow the assumption to preferentially use triangular meshed models for VC, and to limit the quantity of triangles as much as is reasonable. Geometrical objects to be checked against each other with collision detection should especially contain as few triangles as possible. A decreasing quantity of triangles results in an increasing deviation from the exact geometry (Fig. 5-16).

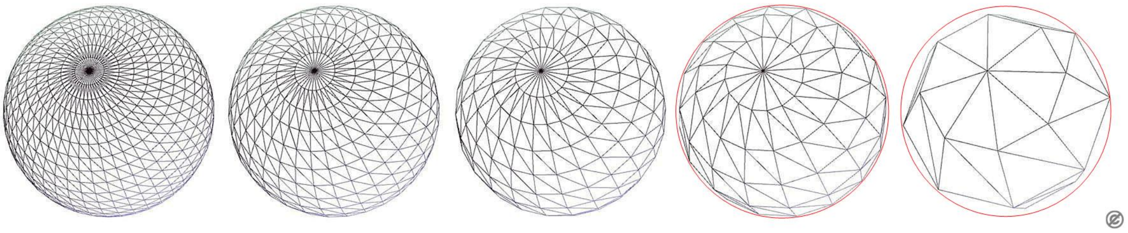


Fig. 5-16: Sphere tessellated into meshes with decreasing quantity of triangles

Strahilov et al. (2012) state a maximum deviation of 2 mm from the exact geometry as being acceptable for their considered use case of robot simulation during VC.

Mesh simplification is a separate area of research for a long time and extensive studying originated a lot of techniques and algorithm (Qiu et al., 2004, Shuming, 2008, Mocanu et al., 2011, Zhou & Wang, 2012, Yong & Hongbin, 2010). Basically, mesh simplification methods can be divided into two groups (Kwak et al., 2010):

- Iterative coarsening of the complete mesh by removing polygons until a specified goal (e.g. number of polygons) is achieved
- Iterative refinement of a newly generated mesh (based on an initial approximation) by inserting polygons

In (Quadros & Owen, 2009) the authors propose to conduct the feature suppression before the generation of a polygonal mesh model because of better results regarding complexity of resulting mesh and because mesh algorithms can fail on features such as holes, fillets chamfers etc. The authors emphasize the use of parameter-based feature suppression available in 3D CAD tools such as SolidWorks or Pro/Engineer prior to mesh generation instead of using design models containing features irrelevant for simulation as the basis for mesh generation, which is often done today. In the case of such preferential approach not being applicable because only models without feature information, B-Reps (Boundary Representation) or NURBS (Non-uniform rational B-Splines), are available to following users, those *"usually spend significant amount of time in manually identifying unwanted features and applying appropriate modification operations to simplify the model prior to mesh generation"*. Thus, Quadros and Owen present an approach for defeaturing based on B-Rep models in such a case.

A comprehensive survey of different simplification techniques applicable to CAD models in form of polygonal meshes or B-Reps specifying simplified kind of features as well as advantages and limitations of methods is given in (Thakur et al., 2009).

According to Kwak et al. (2010), two types of geometric models are used for the simulation of manufacturing systems: freeform surface models usually applied to manufactured products (e.g. in automotive industry) whereas components and sub-systems of production devices are mostly represented by prismatic solid models. They assume that generally also the prismatic solid models are available in the form of triangular meshes. Besides the application of standard simplification procedures of removing local features (see above), the authors state the problem that standard procedures cannot automatically remove invisible internal objects, which they suppose to be *"not necessary for a three-dimensional (3D) plant simulation"*. They propose an approach to remove such unnecessary internal objects (e.g. ball bearings) from mesh models.

Those hidden, invisible parts are not necessarily dispensable for simulation as demonstrated in (Kanai et al., 2012). The visibility change of moveable parts (e.g. the piston of a cylinder) must not be ignored in simplification process, and therefore the authors present an approach that analyzes possible assembly configurations (e.g. piston retracted and piston extended) before starting simplification.

Modern 3D CAD tools already provide several automated simplification features with adjustable filters to remove irrelevant geometrical features and irrelevant geometrical objects. If the functionality of used CAD tool is not sufficient, it is possible to utilise specialised simplification and/or conversion tools such as CADdoctor (Elysium, 2015), CADfix (ITI-Transcendata, 2015) or Nugraf/ PolyTrans (Okino, 2015).

5.3.4.6 CAD Data Preparation - Component structuring

If the CAD data provided are not appropriately structured especially, with separation of moving parts (e.g. actuators), the CAD model is not directly usable for the following steps of Low-level Component Modelling, nor is it possible to base a simulation on such model. In worst case, a CAD redesign may become necessary to provide in the geometry data the separate structures required for simulation.

Hence, adjusting the CAD model inner structure is the crucial factor for CAD preparation (Hoffmann et al., 2012). The model has necessarily to be separated to:

- Static units
- Moving units (actuators)
- Sensors and communication units (transceivers / transponders)

Assemblies created in mechanical engineering are often static, or all parts of an assembly move together. Converting such assemblies to single parts makes sense to minimise the computational power needs of simulation. Sensors have to be separate parts, regardless of belonging to a static unit or a moving unit.

While working with CIROS this method has also proven advantageous for better handling of imported CAD models during geometrical/functional component modelling, because considerably fewer objects appear in the model editor after import; this has been used for e.g. the entire curves or static parts of turnouts (with base frames).

Up to now, the building of assemblies in mechanical engineering is occasionally not done considering the functional interaction of parts, but rather aspects like common manufacturing are respected. Sometimes, that will lead to assemblies wherein static parts are mixed with moving parts. This problem was reported e.g. in (Hollander & Sappei, 2011).

The CAD tool Pro/E respectively Creo Parametric provides a powerful feature called “Shrinkwrap” (University-Cambridge, 2014) that combines simplification with the possible merging of an assembly to a part. To reduce the calculation load as well as for better handling of imported CAD models in CIROS (less objects) the conversion from assemblies to parts can be used were applicable, but one has to pay attention to the structuring requirements stressed above. Hollander and Sappei point out that it is possible to select assembly components to be included in the part to be generated by Shrinkwrap, but by default, all components of the assembly will be merged.

The author’s conclusion is that the CAD preparation for simulation models should be integrated in the mechanical engineering CAD workflow. A concurrent design of CAD models suitable for VC, conducted by CAD specialists, would be better than starting the simulation model building with a CAD preparation of possibly inappropriate CAD models.

5.3.4.7 CAD guidelines

Following guidelines for a CAD design, which meets the requirements of simulation with respect to detailing and model structure, would allow an ordinary straightforward generation of additional 3D-CAD models suitable for VC during mechanical engineering (cf. Fig. 5-20).

Detailing

It is recommended to use the features mentioned above in a parsimonious way, especially radiuses etc. are better used with low detailing resulting in a few facets only. Small features and small objects without functional relevance can be omitted; bigger norm parts (screws, nuts etc.) can be replaced by simplified geometries. The invisible inner structures of objects should not be modelled at all, unless inner parts of a component move outside during operation (e.g. the piston of a cylinder).

Labels and logos should not generate many facets, so it is better to use textures that are defined as separate objects, which can be deleted easily to allow further shortening of calculation time.

Beneficial for simulation model building would be CAD data providing different LOD (levels of detail). CIROS for example supports different LOD that can be changed in simulation.

Model structure

CAD models intended for simulation have to be separated into static objects, moving objects and sensors/communication units. The object tree inside CAD should be modelled in reasonable hierarchical manner, considering rigid bodies forming assemblies being static or moving all together. Moving parts and static parts must not be in the same assembly. Assemblies meeting this requirement can be merged to single parts, because then a later change or restructuring in simulation system is not necessary (and would be difficult and laborious or even impossible). All elements in the object tree should be labelled with descriptive names, not numbers, equipment-specific IDs or cryptic abbreviations, because this is useful for later low-level modelling.

5.3.4.8 Low-level Component Modelling

After CAD preparation and CAD data transfer is completed, the mechatronic component model building is continued with step 4, the main part named Low-level Component Modelling. For the detailed test of control programs impact on the 3D mechanical behaviour of the manufacturing system in VC in an integrated manner, the modelling and simulation of the complete functional chains is necessary.

These functional chains link the user interactions via HMI and control programs through sensors, actuators and drives onto the mechanical movements, and include simulation of mechanical behaviour as well as emulation/simulation of control programs and HMI. The simplified functional chains for the VC of the transportation system are pictured in figure 5-10. To achieve this, it is necessary to build a comprehensive mechatronic plant model composed of mechatronic component models, based on the overall model conceptualisation. The component models should have their conceptual origin already modelled in CAD.

The Low-level Component Modelling, requiring the specification of the functional chains of each mechatronic component model, can be divided into three stages: geometrical, functional and electrical modelling. This non-trivial procedure requires considerable modelling expertise and effort. Especially SMEs usually do not have the modelling experts for this task (cf. chapter 2).

Geometrical Modelling

The geometrical modelling starts with the import of geometry from 3D CAD tools, which is supported by current 3D simulation tools with import filters for native CAD formats or exchange formats. CIROS for example supports STEP (AP203/214), STL, VRML and IGES. If the simplification of exceedingly complex geometric data and model restructuring had not been done in a preceding CAD preparation inside a 3D CAD tool, it would become necessary to make up leeway. Another reason could be the CAD data transfer, which has produced an unsuitable structure in the geometrical model inside the simulation tool.

In principle, the preparation of CAD data can be done inside the simulation tool as indicated in figure 5-8, which is actually done if supported by simulation tool, but then dispensable data has to be deleted and the model has to be restructured there with cumbersome detail work. CIROS for example provides since version 5.1.4 used at the UASA Hannover the merging, aligning and optimisation of CAD data by e.g. replacement of selected polyhedrons by geometric primitives (cylinder, cuboid and sphere), closing holes and reducing number of facets by mesh simplification and simplification of roundings and cylinders.

It would be more efficient and faster overall to prepare the 3D CAD data within the 3D CAD tool, or deploy specialised simplification and/or conversion tools. In comparison with these tools, the relevant simplification capabilities of simulation tools are limited and less controllable and the restructuring work is associated with more effort. This would be especially beneficial in a situation where the simulation tool has to be used by e.g. a control engineering specialist, because a simulation specialist skilled in 3D modelling is not available, as is usually the case in SMEs. Then the 3D modelling has to be done by 3D CAD specialists using their well-known CAD tools for this purpose.

Adjusting the inner structure of geometrical models considering static parts, actuating elements and sensors is the crucial step in creating a model ready for the following functional modelling. An example for an unsuitably structured geometrical model after importing standard CAD data is pictured in figure 5-17. The unstructured CAD model of the turnout component of the transportation system has been transferred from SolidWorks to CIROS (①), resulting in the same long list of single objects. In such a case, a manual hierarchical restructuring (②) of the CAD model into objects, sections and components or hulls is recommended to provide a well-structured model for the following functional modelling. The goal of restructuring is to have as few as possible objects or groups, where each rigid body should form a group, and kinematic degrees of freedom can only defined between groups of the same object.

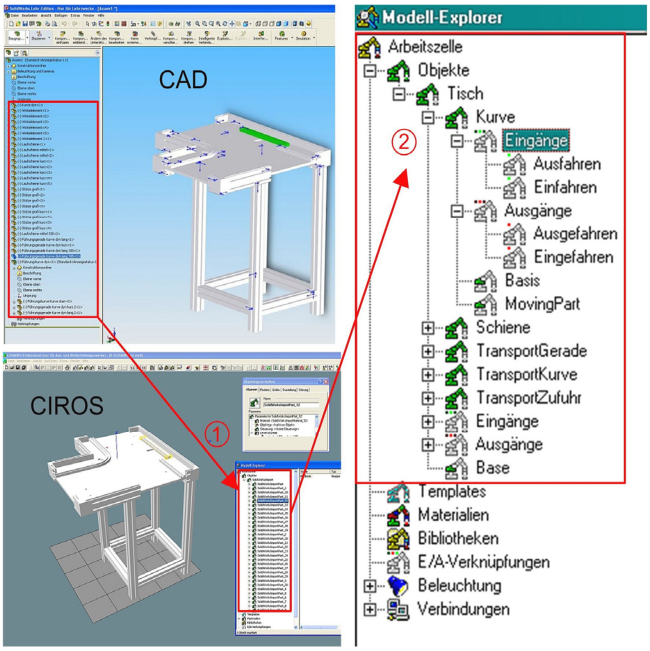


Fig. 5-17: Import of unsuitable structured CAD data to CIROS

In the worst case, e.g. static and moving objects merged to a single part in CAD model, a CAD redesign or an attempt of a laborious clearance cutting inside CIROS may become necessary to provide the necessary structures in the geometrical model. During the course of the reported project, suitable modelling has been considered through the 3D CAD modelling of the transportation system components.

Functional Modelling

In the vast majority of cases, it is not possible to transfer kinematics from CAD tools to simulation tools. Either the import of native CAD formats which support kinematics or movement restrictions alone in simulation tools is not possible. The principal transfer possibility of exchange formats such as STEP or COLLADA has not yet been implemented in tools (see above). Therefore, it is generally necessary to manually equip the geometrical models with kinematics and where applicable with sensor functions and/or communication functions. Besides the specification of motion constraints and the degrees of freedom, the position and effective direction of sensors has to be defined.

CIROS provides several actuator mechanisms (e.g. translation, rotation, gripping), different types of sensor functions (e.g. ultrasonic, optical, capacitive, inductive) and transceiver/transponder functions for this purpose in the library (cf. section 5.2.1). These functions are allocated manually by using the invisible mechanisms to the respective parts of the geometrical model, which results in the definition of an integrated functional model containing for example push cylinders and sensors as e.g. the track turnout of the transportation system. Both two-way push cylinder with limit sensors and the three track sensors are modelled this way. These simulated actuators and sensors need parameterisation e.g. stroke and speed has to be applied to the moving piston of a cylinder or timing, measurement range, switch distance and hysteresis of sensors. Therefore, it is necessary to have the needed technical data available for simulation model building, where these data have to be manually added to the functional models. If the use of different physical measurement principles of sensors is intended for VC, it is necessary to define e.g. material and colour for objects to be detected in this stage. For the use of physical simulation with e.g. Nvidia PhysX, additional parameterisation would become necessary because data for e.g. mass and friction are needed.

Electrical Modelling

For the final Low-level Component Modelling task, i.e. electrical modelling, one has to manually assign electrical inputs and outputs to sensors and actuators in the functional models. CIROS allows interactive, graphical editing of these connections. These I/Os will be linked to I/Os of control programs to specify the mapping between actuators and sensors of virtual manufacturing system components with (virtual) controller I/O's for VC which results in closed functional chains. The simulation of "intelligent" components such as the transport car with built-in microcontroller is possible by implementing a logic controller into the functional model. This logic controller provides e.g. basic calculating operations, Boolean logic (NOT, AND, OR, XOR, NAND, NOR) and flip-flops. If this functionality is not sufficient, it is also possible to couple the functional model with an internal virtual controller programmed with IRL.

5.3.4.9 Verification and Validation of mechatronic component models

The final task for the modelling and simulation expert is V&V as part of step 4, and completes the creation of the mechatronic component model that can be stored in a library for later High-level Plant Modelling. The verification has to detect incorrect implementation of a component model (based on concept, specification and data from steps 2 and 3) arising from errors during geometrical, functional or electrical modelling. Such errors can be quickly corrected within the modelling tool, remaining in step 4. Errors detected by validation lead to a review of conceptualisation, specification and data from steps 2 and 3; maybe assumptions made during conceptualisation are false. Ideally, it is possible to validate the simulation model by comparing it with the real component.

5.3.4.10 Storage in Simulation Model Library

After validation, the mechatronic component model is ready for repeated (re-)use and can be stored in a simulation model library in step 5. When all component models have been stored in the library, the modelling expert is no longer required and a simulation user is able to conduct the High-level Plant Modelling and finally the VC. It is recommended to support these users with a brief documentation on the component model and its use.

5.3.5 High-level Plant Modelling and VC

A simulation user (e.g. a trained control engineer/commissioning engineer) will now be able to master the High-level Plant Modelling and to conduct the VC at the end, following the proposed procedure in figure 5-18.

5.3.5.1 Plant Model Definition and Conceptualisation

In step 1, the user should determine the subsystems, stations and components that have to be part of the plant simulation model, depending on its intended purpose. The specification of needed parameters, variables as well as relationship between selected objects for layout and interconnection completes step 1.

5.3.5.2 Data Collection and Data Preparation

After identification of all component models, it is necessary to acquire variable/adjustable component parameters to be used in the real system. Examples are adjustable measurement range of real sensors, parametrizable speeds of transport cars etc. Besides layout data, the I/O lists are indispensable for later connection of the (virtual) controller to simulation model.

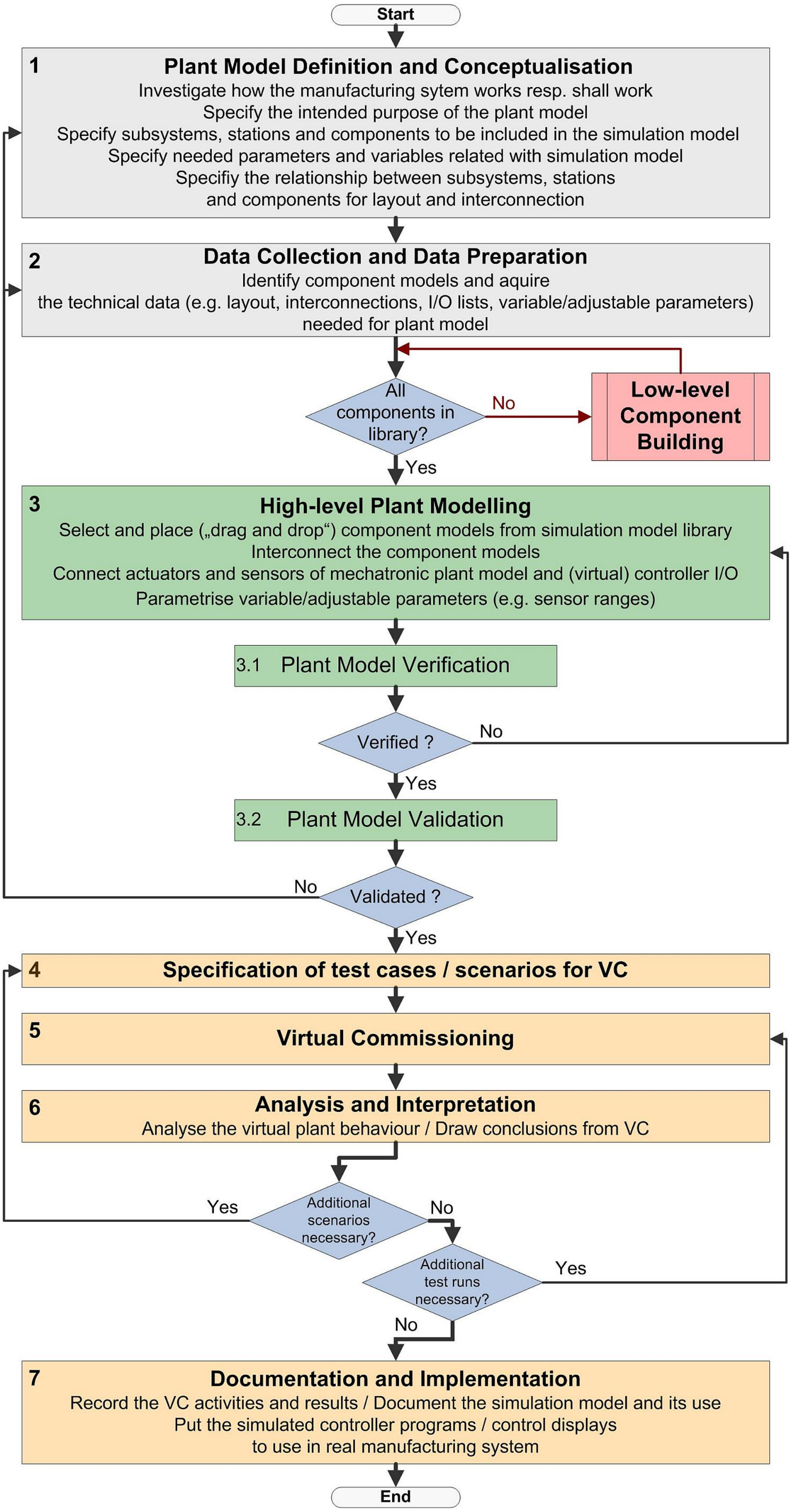

Start
1
Plant Model Definition and Conceptualisation
Investigate how the manufacturing sytem works resp. shall work
Specify the intended purpose of the plant model
Specify subsystems, stations and components to be included in the simulation model
Specify needed parameters and variables related with simulation model
Specifiy the relationship between subsystems, stations
and components for layout and interconnection
2
Data Collection and Data Preparation
Identify component models and aquire
the technical data (e.g. layout, interconnections, I/O lists, variable/adjustable parameters)
needed for plant model
All components in library?
No
Low-level Component Building
Yes
3
High-level Plant Modelling
Select and place („drag and drop“) component models from simulation model library
Interconnect the component models
Connect actuators and sensors of mechatronic plant model and (virtual) controller I/O
Parametrise variable/adjustable parameters (e.g. sensor ranges)
3.1 Plant Model Verification
Verified ?
No
Yes
3.2 Plant Model Validation
No
Validated ?
Yes
4
Specification of test cases / scenarios for VC
5
Virtual Commissioning
6
Analysis and Interpretation
Analyse the virtual plant behaviour / Draw conclusions from VC
Yes
Additional scenarios necessary?
No
Additional test runs necessary?
Yes
No
7
Documentation and Implementation
Record the VC activities and results / Document the simulation model and its use
Put the simulated controller programs / control displays
to use in real manufacturing system
End

Fig. 5-18: Proposed procedure for systematic High-level Plant Modelling and VC

5.3.5.3 High-level Plant Modelling

If all component models are in the library, the first task of step 3 (Fig. 5-18) is to select all required components from the library and to place and interconnect them with the plant model editor to the virtual manufacturing system. Subsequent the data acquired in step 2 has to be applied to adjustable parameters such as the sensor ranges. Finally, the actuators and sensors of the virtual plant model have to be connected to the (virtual) controllers based on the I/O list.

5.3.5.4 Verification and Validation of Plant Model

Since every component model has run through a V&V procedure before it has been stored in the simulation model library, the V&V, especially the verification, of the plant model should be easier. The V&V as part of step 3 is the final task of High-level Plant Modelling.

The verification, carried out manually, has to detect any incorrect implementation of the plant model concerning interconnection and parameterisation of component models as well as faulty connection of plant model and controllers. Typical errors detected by verification are e.g. false parameterised variable sensor measurement ranges or wiring errors when connecting actuators and sensors of the virtual plant model with controller I/Os. Such errors can be quickly corrected staying within step 3, by means of the modelling editor in the simulation tool.

The validation has to give assurance that the plant model correctly represents the real world regarding its intended use by appropriate system assumptions and correctly conducted modelling (cf. subsection 2.3.4). The validation of the entire manufacturing system has to be done thoroughly although the component models have been validated individually to preclude errors resulting from interactions between components in unforeseen situations. If all potentially erroneous situations of the entire system with its interacting components have been considered during modelling, the simulation model is able to react to errors of controller programs in an adequate manner and the VC analyst will be able to recognize errors of control programs by reactions of the simulation model. Errors detected by validation trace back to plant model conceptualisation and data from steps 1 and 2; maybe assumptions supposed to be correct are in fact false or data collected are incorrect. It is recommended to call in a domain expert (e.g. system operator) for this task. Ideally, the plant simulation model can be validated by comparing it with the real manufacturing system.

5.3.5.5 Specification of test cases for VC

For the VC studies in this thesis, the test cases have to be designed manually (step 4) as it is usually the case in industrial applications of VC. In the first step of the proposed general methodology (Fig.

3-1), the goal of the simulation study has to be defined and the questions to be answered by VC have to be predefined. The outcome of this is the definition of the plant models' intended purpose. From this, the specification of test cases follows. Besides the valid straightforward test cases for V&V of normal control program operation, one advantage of VC is the possibility of risk-free simulation of errors cases, up to worst-case scenarios. The consideration of all potentially erroneous situations of the manufacturing system poses a challenge for the specification of test cases; the implied effort becomes not only necessary for VC, but is the same as for real commissioning and certainly for the prior control programming.

The automated test case generation and automated testing are particular fields of ongoing research in the periphery of VC research (Kabitzsch et al., 2008, Hametner et al., 2011, Naake et al., 2012, Greifeneder & Gohr, 2014, Ramler et al., 2014, Magnus et al., 2015, Thron et al., 2016), but has not been considered in depth in this thesis.

5.3.5.6 Virtual Commissioning, Analysis and Interpretation

The Virtual Commissioning starts with the opening of the mechatronic plant model within the simulation tool, downloading all controller programs from associated IDEs into the real or virtual controllers (HIL or SIL configuration) involved and loading the HMI/SCADA application. The execution of VC has similarities to the execution of real commissioning. After starting all applications/systems, the defined test cases for all operating modes can be executed.

First, normal operating modes such as initialisation, start-up, manual operation and automatic mode(s) including safety functions/interlocking implemented in control programs are tested, followed by special functions/programs and error scenarios. In manual mode, the VC analyst (control programmer/commissioning engineer) can activate single actuators via the control display and supervise the moving of actuators in the 3D simulation model. In doing so, the correct moving direction and the faultless activation and signalling of end position sensors is checked. At this stage, all interlocks that are supposed to be implemented in control programs should be tested. In mechatronic systems, it is often required to prevent the movement of actuators depending on the position of other actuators or the activation has to be conducted in the correct order to prevent collisions.

For the transportation system considered in this thesis, it is necessary to avoid e.g. the opening of stopper units if the following turnout is in the wrong position because the transport car would crash (cf. chapter 7), in manual mode as well as in automatic programs. Not only should the interlocks necessary for separate subsystems be tested, but also the cooperation of different subsystems (e.g. correct exchange of interlock and enable signals) such as transportation systems and robots.

The successful testing of manual mode is followed by the testing of automatic modes. If the steps of automatic programs are always conducted in the same order, one test run for each program should be sufficient. However, if variations are possible, several test runs might be necessary.

The preceding tests are intended to find errors in control programming, projected wiring and as the case may be in mechanical set-up, but also the process itself with its implemented components can produce errors. Faults are possible by e.g. defective sensors (no signalling), actuators not moving when activated (e.g. by compressed-air failure) or workpieces falling down (e.g. from carrier of transportation system or inside robot cell). All these can be part of defined error scenarios to be tested during VC, up to worst-case scenarios that would result in damages to equipment or endangering operating personnel and therefore not tested during real commissioning. If such tests are conducted during VC, this would allow the checking if control programs are able to detect such errors (e.g. by time monitoring between activating actuator and feedback signal of end position sensor) and if the programs respond adequately to the occurrence of failures. CIROS for example supports the testing of different failure scenarios through a GUI to set failures or to define test sequences. Besides debugging of control program errors or design errors, the analysis of virtual plant behaviour provides the opportunity to rethink whether the principal control strategy of the entire system is reasonable, even if no other actual bugs occur. Besides these, additional and possibly deleterious situations might be identified when viewing the running 3D simulation.

5.3.5.7 Documentation of VC and Implementation

All executed test scenarios and results (e.g. detected errors) should be documented, as should the applicability of the simulation model regarding the range of uses, limitations and drawbacks. Detailed documentation will be useful if the simulation model is to be reused or reported, e.g. in the case of redesign of the manufacturing system. The potential reuse of the simulation model would require that the model and its documentation should be kept up-to-date, all later changes to the real system should be reproduced in the simulation model. The implementation of controller programs and control displays facilitate the real commissioning. Errors during real commissioning should be analysed. Not all errors, e.g. wiring errors by the electrician or mechanically misaligned sensors, can be detected by VC, but in other cases one should reflect on why the error had not been found during VC (possibly due to errors or limitations of simulation model).

5.4 Summary of procedural steps for VC of manufacturing systems

The following tables summarise the procedural steps including activities and/or questions to be answered for planning, implementation and execution of VC. The tables provide more details than the presented flow-charts (Figs. 3-1, 5-9, 5-18) to clarify the stages of proposed methodology. The tables are separated into the three main phases (conception, implementation and analysis) introduced in chapter 3.

Procedural step	Tasks: activities to be conducted and/or questions to be answered
Problem Definition and Cost Analysis	• Define the goals of the VC study such as: - Meet the deadline for SOP or even put ahead the SOP - Better quality of automation software before real commissioning - Reduce time and costs for real commissioning - Save hardware and erection work for test purposes - Minimize downtimes for future retrofit or extension of the plant - Continue simulation after VC for e.g. operator training • What questions are to be answered by VC? - Check of manufacturing system design and functionality or only V&V of automation software? - Which aspects of manufacturing system design/functionality and/or automation software (control software/HMI/SCADA) are to be tested? - Simulation of operator errors and/or defective components? - Simulation of critical manufacturing system status (not tested during real commissioning due to risk of damages/endangering people)? => Document: Goal definition • Is VC the right method in consideration of expected cost/benefit ratio? - Do the complexity of analysis task and the need for certainty justify the use of VC? - Conduct an effort-benefit analysis taking complexity and rentability into account and consider to divide a complex overall VC task into separate VC sub-projects
Project Planning	• Ensure to have sufficient time, personnel with required knowledge and skills (modelling/simulation) and necessary HW/SW available - How much time is available? Timely realisation considered as being realistic? - Existing dependencies (temporal/organisational) to other projects? => Definition of project time schedule - Who has the necessary knowledge/skills about automation HW/SW, process/plant, real commissioning and modelling/simulation? => Definition of project team / project manager - What to do if modelling/simulation skills are missing? => Hiring, training, consulting… => Document: Project master plan • Analyse manufacturing system regarding principal design (stations, subsystems), material flow, process sequences, automation HW/SW (controller, communication infrastructure, programming languages), interfaces • Define principal test cases/scenarios based on goal definition (see above) • Specify principal model types (see subsection 2.6) and real-time conditions

	• Specify VC environment (cf. Fig. 5-5) - Select suitable 3D plant simulation tool - Select simulation method (HIL/SIL) and communication infrastructure for simulation (OPC,…) => Document: Requirement specification
Plant Model Definition and Conceptualisation	• Investigate in detail how the manufacturing system works or shall work • Specify the intended purpose of the plant simulation model (based on requirement specification) • Divide the manufacturing system into subsystems, station and components - First stage of decomposition: Subdivide the entire system into possible separate VC projects - Second stage: Further decomposition to support a modular, component-based simulation model building approach • Determine subsystem, stations and components to be included in the plant simulation model (depending on its intended purpose) • Specify needed data (parameters, variables …) related with components relevant for simulation such as: - Sensors: Physical measurement principal, timing, measurement range, switch distance, hysteresis etc. - Actuators: Speed, stroke etc. • Specify relationship between subsystems, stations and components for layout, interconnection and conceptualisation • Conceptualisation of simulation model for entire manufacturing system with functional relationship (functional chains) between subsystems/components (cf. Fig. 5-10)
Data Collection and Data Preparation	• Identify, acquire, prepare and adapt the data needed for simulation model of the manufacturing system - Simulation models of subsystems/components if available in simulation model library of selected simulation tool - 3D CAD models of components/subsystems not in library - Behaviour models/descriptions of components/subsystems not in library - Interconnection data (detailed layout of manufacturing system and detailed lists of I/O signals) - Technical data of sensors/actuators - Definition of additional functionalities/signals (e.g. emergency stop switches) to be considered in VC => Data requirements for simulation model building - Controller programs for PLCs and robots - Control displays for HMI/SCADA systems => Data requirements for VC

Table 5-2: Conception phase

If all needed models are in the simulation model library, the simulation model building comprises only the High-level Plant Modelling (Table 5-3); otherwise, a Low-level Component Building (Table 5-4) becomes necessary.

Procedural step	**Tasks: activities to be conducted and/or questions to be answered**
High-level Plant Modelling	• Select, place and interconnect component models from simulation model library • Connect actuators/sensors of mechatronic plant model with I/Os of (virtual) controllers • Parametrise variable/adjustable parameters of components
Plant Model Verification	• Detect incorrect implementation of plant model - Interconnection and parameterisation of component models - Connection of actuators/sensors of plant model with controller I/Os => Correction inside modelling editor of simulation tool
Plant Model Validation	• Ensure the correct representation of the real world (real manufacturing system) by the simulation model (mechatronic plant model) regarding its intended use (Table 5-2, Goal definition) - Has to be done thoroughly although component models have been validated before storage in library (Table 5-4) to preclude errors resulting from interaction of components - It is recommended to call in a domain expert (e.g. system operator) => Detected errors or discrepancies trace back to plant model definition/conceptualisation or collected/prepared data

Table 5-3: Implementation phase – High-level Plant Modelling

The different stages of 3D models from standard CAD models for mechanical engineering to validated Mechatronic Component Models are indicated by ❶ - ❽.

Procedural step	**Tasks: activities to be conducted and/or questions to be answered**
Component Definition	• Define components supposed to be relevant for VC (based on the decomposition of the manufacturing system)
Component Model Conceptualisation	• Investigate how the component works or shall work. Explore component functionality as part of the entire system • Define component boundaries, restrictions and specify interfaces to other components • Determine component detailing and specify required properties and parameters • Validate conceptual model if appropriate. Develop a conceptual behaviour model, e.g. as Function Block Diagram (Fig. 6-15), for components

	with complex logics/behaviour (such as the transport car in this thesis). Such models can be validated using a PLC IDE (Fig. 6-16)
Data Collection CAD Data Preparation and CAD Data Transfer	• Identify, acquire, prepare and adapt data needed for components - Technical data (e.g. speed of actuators, measurement range of sensors) - Behavioural models/descriptions if available - ❶ 3D CAD models for mechanical engineering (available from component manufacturer, web portals / In-house CAD development) • Prepare CAD data for simulation - Component Structuring: Structure CAD models considering - Static units vs. moving units (actuators) - Sensor / communication units (e.g. transceivers/transponders) - Simplification: Simplify CAD models to allow for real-time simulation, 3D rendering and visualisation. Operational simplification tasks: - Object filtering: Complete removing or (automated) replacing with simplified geometries (hidden invisible parts, small objects or other selected parts) - Feature suppression: Reduce detailing or completely remove geometrical features irrelevant for VC (e.g. holes, bosses, chamfers, steps , labels, logos…) - Mesh simplification: Reduce quantity of triangles => ❷ Simplified and adequate structured CAD models suitable for VC CAD data preparation can be conducted in CAD tools and/or in simulation tools • Transfer CAD data from CAD tool to simulation tool => ❸ Exported CAD models in exchange format (e.g. STEP) or native
Low-level Component Modelling	• Import and review (possibly pre-process) CAD data => ❹ Component models in native format of simulation tool • Geometrical modelling => ❺ Appropriate structured Geometrical Component Model • Functional modelling => ❻ Functional Component Model (FCM) with kinematics/actuators and sensors • Electrical modelling => ❼ Mechatronic Component Model (FCM with assigned electrical I/Os and logics/behaviour (if applicable)
Component Model Verification	• Detect possibly incorrect implementation of a component model (based on conceptualisation, specification and data) arising from errors during Low-level Component Modelling => Correction inside modelling editor of simulation tool
Component Model Validation	• Ensure the correct representation of the real component by the simulation model of the component (ideal would be the comparison of real and virtual component) => Detected errors or discrepancies trace back to component model definition/conceptualisation or collected/prepared data => ❽ Validated Mechatronic Component Model

Simulation Model Library	• Store the validated Mechatronic Component Model in the simulation model library - Support the users with a brief documentation on the component model and its use

Table 5-4: Implementation phase – Low-level Component Building

The last table summarises the procedural steps of the analysis phase.

Procedural step	**Tasks: activities to be conducted and/or questions to be answered**
Experimental Design of VC	• Detailed specification of test cases/scenarios for VC - Usually manual specification of test cases/scenarios based on principal specification/goal definition (see table 5-2) - Several test runs for particular scenarios recommended (e.g. if variable execution of control programs is possible)? - Consider automated test case generation/automated testing if supported by simulation tool
Virtual Commissioning	• Conduct the VC simulation study - Load mechatronic plant model inside simulation tool, download all controller programs from associated IDEs into real/virtual controllers (HIL/SIL), load HMI/SCADA applications and start all systems - Check coupling of simulation tool and controllers and the correct mapping of I/O signals - Start with test of single controllers and subsystems, then test the cooperation of different controllers/subsystems and finally the VC of the entire manufacturing system - Test manual modes by activating single actuators and supervising correct movements and signalling (end positions etc.) - Test safety functions/interlocking in manual mode - Test other modes such as initialisation, start-up and automatic mode(s) - Test error scenarios (e.g. defective components, operator misuse) up to worst-case scenarios
Analysis and Interpretation	• Analyse the virtual plant behaviour / Draw conclusion from test runs - Additional scenarios necessary? If not all predefined questions are addressed by current test cases or additional scenarios (e.g. possibly deleterious situations) have been identified during VC, new test cases have to be designed - Additional test runs necessary? Control programs with variable order of subtasks might require repeated execution of the same test case as well as corrected errors in control programs
Documentation and Implementation	• Record the VC activities and results - Document all executed test scenarios, repeated test runs and their results (e.g. detected errors) • Document the plant simulation model and its use - Applicability regarding the range of uses - Limitations and drawbacks => Document: VC documentation

	• Put the tested and corrected controller programs/control displays to use in the real manufacturing system and conduct the real commissioning - Not all errors can be detected by VC (e.g. wiring errors/misaligned sensors) – analyse other errors during real commissioning on why they had not been found by VC (errors/limitations of simulation model?)

Table 5-5: Analysis phase

5.5 Proposal for novel Low-level Component Modelling workflow

At present the same component or sub-system has to be modelled conducting an extensive CAD preparation and complex Low-level Component Modelling as described above again and again for different simulation tools, but also for the same simulation tool at different users i.e. manufacturing system builders, because generally a model exchange does not exist. Hence, the workflow for VC outlined in figure 5-19 has always to be repeated for different simulation tools and different users building a mechatronic plant model based on the same components.

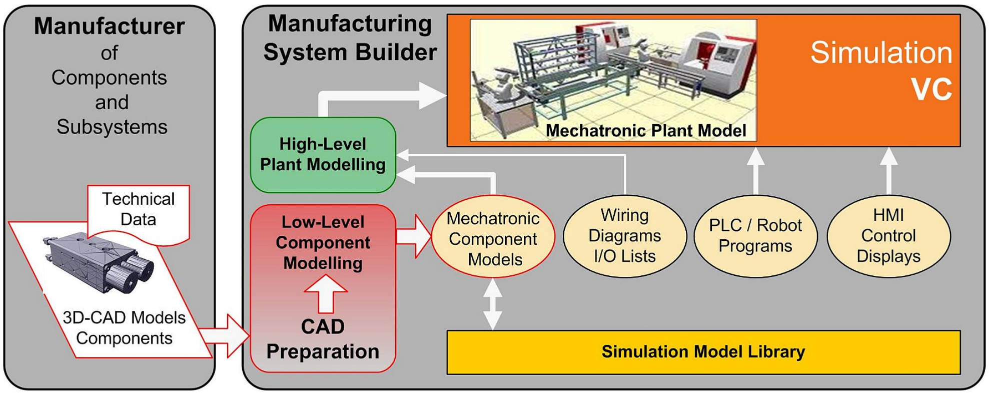

Manufacturer of Components and Subsystems
Technical Data
3D-CAD Models Components
Manufacturing System Builder
Mechatronic Plant Model
Simulation VC
High-Level Plant Modelling
Low-Level Component Modelling
CAD Preparation
Mechatronic Component Models
Wiring Diagrams I/O Lists
PLC / Robot Programs
HMI Control Displays
Simulation Model Library

Fig. 5-19: Workflow for VC with conventional Low-level Component Modelling

Two key aspects for improvement arise out of the present situation to reduce the low-level modelling effort for the simulation user:

1. An enhanced approach for Low-level Component Modelling based on CAD data
2. The creation of mechatronic component models by the manufacturers for the future

The first aspect is supported by a Low-level Component Modelling procedure (including CAD preparation) by simulation users following a systematic workflow as proposed and tested in this thesis.

Further improvements would result from a relocation of CAD preparation for VC following CAD guidelines presented above (cf. subsection 5.3.4.7), to the component manufacturers, as a first step.

From the author’s point of view, it is reasonable and essential to relocate the low-level modelling procedure as far as possible to the origin of products, i.e. to the designers, manufacturers and suppliers of components and sub-systems, and finally the manufacturers should provide mechatronic component models, with commercial competitiveness being the driver for their supply.

5.5.1 Start of CAD preparation for VC during CAD design for mechanical engineering

An important first step towards simulation model building for VC to become a task for control system engineers and commissioning engineers from SMEs, would be the unimpeded transfer of CAD data to component modelling in simulation tools. The data flow should facilitate and be directly applicable to Low-level Component Modelling suitable for VC. Therefore, it is required that CAD designers in companies producing off-the-shelf components and sub-systems and designers of purpose-built components and sub-systems e.g. from plant manufacturing companies and their subcontractors consider the supply of such data.

The supply of appropriate CAD data would greatly reduce the laborious CAD preparation always repeated by every simulation user to a one-time activity of 3D modelling specialists.

The Low-level Component Modelling that remains still necessary at this stage (Fig. 5-20) would be better facilitated and simplified. To achieve this goal there is a need for information to be gathered from designers creating CAD models, persuasion the manufacturers of components and sub-systems to provide additional CAD data be recommended to facilitate component models for VC. In order to advance thinking in mechatronic units, CAD education in universities should address model transferability, simplification and the reuse for Low-level Component Modelling intended for 3D simulation.

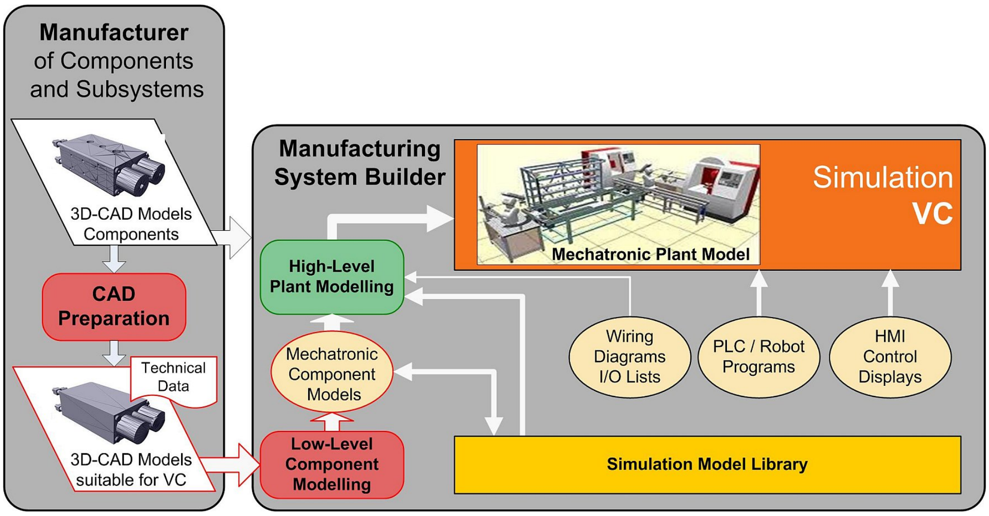

Manufacturer of Components and Subsystems
3D-CAD Models Components
CAD Preparation
Technical Data
3D-CAD Models suitable for VC
Manufacturing System Builder
Mechatronic Plant Model
Simulation VC
High-Level Plant Modelling
Mechatronic Component Models
Low-Level Component Modelling
Wiring Diagrams I/O Lists
PLC / Robot Programs
HMI Control Displays
Simulation Model Library

Fig. 5-20: Workflow for VC with low-level modelling supported by CAD data from supplier

The component designers in mechanical engineering have a clear view of the functional structure of engineered components. Today, however, only part of this general functional view is implemented in CAD models, which is represented predominantly by CAD drawings with additional object specifications for mechanical engineering. This means that the additional functional knowledge of the designer must be documented separately or transferred from the designer to simulation tool users in a bespoke manner. If manufacturers of off-the-shelf components could be convinced to provide 3D CAD data suitable for VC, by following CAD guidelines as proposed in subsection 5.3.4.7, this would be a good basis for building and provision of future mechatronic component models.

5.5.2 Proposal for novel model collection methodology

The systematisation of simulation model building based on component CAD data is a main goal of this research project. However, in future, the import of complete mechatronic component models into the internal libraries of different available 3D plant simulation tools with minimum effort would be an advantageous development. Therefore, a new strategy for collecting mechatronic models from component and sub-system vendors, which will allow the provision of such models together with the hardware components, is proposed below.

For simulation systems in electronics (e.g. PSPICE), the component simulation models are often provided by the component manufacturers. So in the long term a similar approach should be taken also for the mechatronic components of manufacturing systems, i.e. the simulation models should be provided by the component manufacturers. This means that the manufacturers (and not the simulation users) will have to handle the low-level modelling (including CAD preparation) of their components.

The vision is the enlargement of simulation model libraries according to requirements of any virtual manufacturing system with no, or only little additional work to allow a VC mainly based on comparatively easy High-level Plant Modelling. If a manufacturing system is assembled only by off-the-shelf components or sub-systems a Low-level Component Modelling should not be necessary any longer. A typical case might be the provision of mechatronic models for most mechatronic components within the broad variety of available transportation systems (cf. chapter 5), which are implemented in many manufacturing systems today. Generally current simulation systems provide only generic models or a small choice of models for real hardware in their libraries, e.g. CIROS for Bosch-Rexroth TS2plus. In the case of implemented purpose-built items, a CAD preparation following the CAD guidelines and a low-level modelling has to be carried out. In order to set-up the enlarged simulation model library a model collection systematic by which the simulation model data can be collected from the component manufacturers, will have to be established. This is in addition to the technical data available as standard. It would not then be necessary any longer to create mechatronic component models with great effort (especially CAD preparation and Low-level Component Modelling), in order to conduct a VC of the manufacturing system.

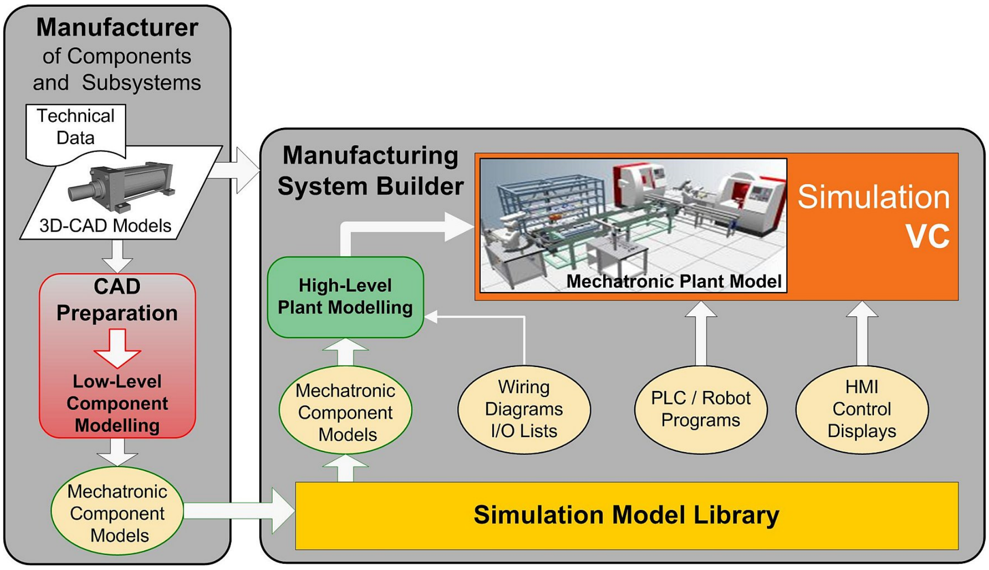


Fig. 5-21: Novel workflow for VC with mechatronic component model provision by supplier

The mechatronic component (or sub-system) models provided by manufacturers (Fig. 5-21) should be tool-independent, because it is not reasonable for the manufacturers to create tool-specific models for many different 3D plant simulation tools. For this purpose, it will be necessary to utilise a standardized structure and data exchange format for mechatronic models in future. Customer request for such models, not only CAD data or electronic datasheets like today, would accelerate this process. Experts from universities, in cooperation with component manufacturers, could make the first move to build up such exemplary model libraries of components or sub-systems. This approach is supported and would initially be enabled by the proposed principal of splitting Low-level Component Modelling and High-level Plant Modelling, based upon modularisation and library use.

5.5.3 Proposal for exchangeable mechatronic component models

The final goal of the new approach (Fig. 5-21) would be the availability of complete mechatronic models incorporating the whole functional chains through sensors, actuators and drives onto the mechanical movements. These models would bear resemblance to the tool-specific mechatronic component models available today at end of Low-level Component Modelling inside simulation tools, but provided by the manufacturers of components and sub-systems in a standardised, tool-independent format.

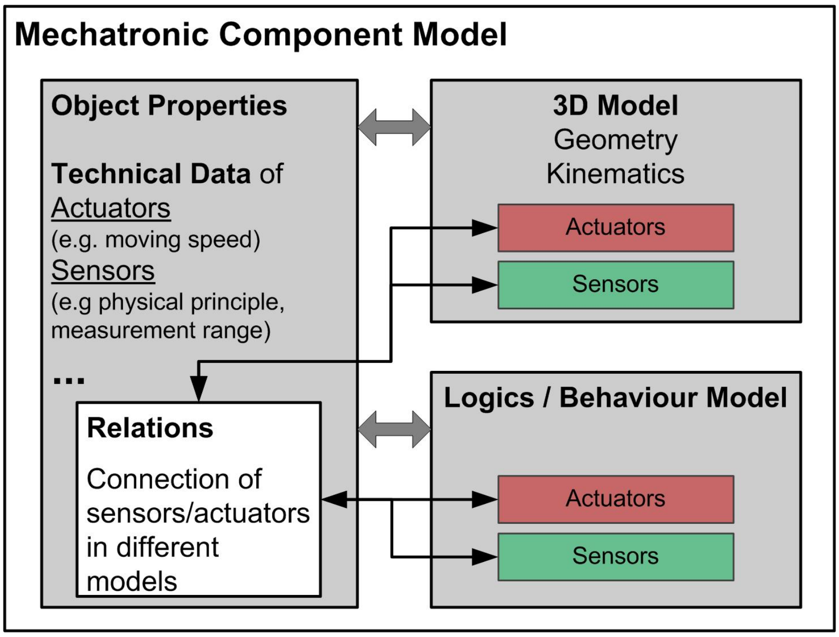


Fig. 5-22: Mechatronic component models for VC

Such a mechatronic component model (Fig. 5-22) has to have the ability to incorporate at least geometry, kinematics, sensors, electrical inputs/outputs, behaviour/logics and technical data for actuators and sensors. Even if 3D models with geometry and kinematics would be available for import into simulation tools, the information which geometrical objects are sensors is still missing, as well as the linkage to a logics description respectively behaviour model. The sensors in the 3D model have to be identifiable (position) but the information of effective measuring direction is also essential for simulation. In addition, the technical data for actuators and sensors relevant to simulation, e.g. motion speed of actuators, physical principle of sensors, measurement range of sensors etc., should be included in such a model. The technical data should be expandable with additional information such as material, mass, friction etc. but also non-technical data, not directly relevant to simulation, such as purchase numbers, documentation should be possible to add.

The proposed provision of tool independent mechatronic component models would be given by the application of standardised exchange formats such as AutomationML for this purpose.

5.5.4 The potential applicability of AutomationML for the proposed new approach

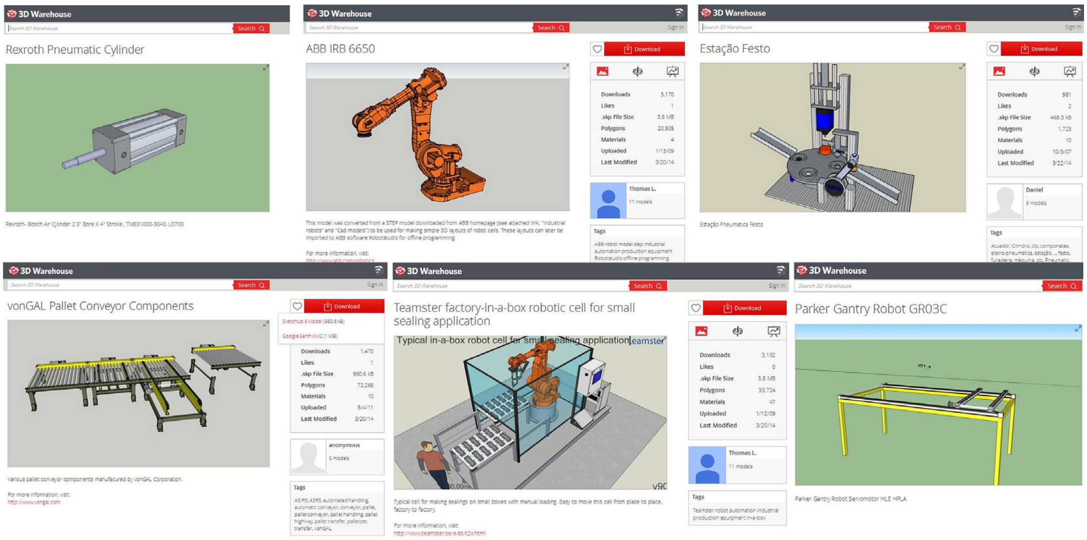
Hitherto, existing approaches incorporating AutomationML (AML), mainly from automotive industry, focus on the use of mechatronic objects to build up mechatronic cell models. As part of the "Digital Factory" concept, this approach follows a *"company-specific strategy"* (Kiefer et al., 2011). The mechatronic models are stored in a *"centrally organized mechatronic resource library"* to allow *"company-wide access"* (see also Fig. 2-9).

These restrictions to company internal use apply also to the “mechatronical units” as part of a “mechatronic engineering process” for manufacturing system engineering defined in (Lüder et al., 2010b), which is also based on such a company specific mechatronic resource library. Lüder and co-workers propose that such a mechatronical unit (MU) should incorporate topological data (internal hierarchy, interfaces), mechanical data (3D CAD, kinematics), electrical and fluidic data (connections, wiring diagrams), process control data (signal information, PLC function blocks), function describing data (controlled/uncontrolled behaviour) and generic data such as technical and organisational data. In their modelling proposal for MUs with AutomationML, the behaviour description stored in a PLCopen file and 3D data, kinematics in a COLLADA file are referenced by special interfaces. A linkage between 3D model and behaviour model has not explicitly been proposed.

Thus, such approaches that are based on company internal mechatronic libraries, only suitable for companies (e.g. from automotive industry) making intensive use of the Digital Factory concept, differ from the procedure proposed herein, which would be beneficial for SMEs too.

The principal applicability of AutomationML for this new model exchange approach is given by incorporating CAEX, COLLADA and PLCopen XML, even though 3D plant simulation tools today mostly do not facilitate these formats.

The exchange of separate COLLADA models is already in existence. The 3D Warehouse (Trimble, 2014), the former Google SketchUp portal, provides models in SketchUp format (.skp) and sometimes in COLLADA format. The models are often packed in a .kmz file, which is a zipped file containing the COLLADA file (.dae). In the meantime, also many industrial parts such as pneumatic cylinders, conveyors, robots or even complete cells are available (Fig. 5-23).



Rexroth Pneumatic Cylinder

Rexroth- Bosch Air Cylinder 2.5" Bore X 4" Stroke , TM831000-3040, L0700

ABB IRB 6650

Downloads	5,170
Likes	1
.skp File Size	3.8 MB
Polygons	20,505
Materials	4
Uploaded	1/13/09
Last Modified	3/20/14

Thomas L.
11 models

This model was converted from a STEP model downloaded from ABB homepage (see attached link, "Industrial robots" and "Cad models") to be used for making simple 3D layouts of robot cells. These layouts can later be imported to ABB software Robotstudio for offline programming.

For more information, visit:
http://www.abb.com/robotics

Tags
ABB robot model step industrial automation production equipment Robotstudio offline programming

Estação Festo

Downloads	961
Likes	2
.skp File Size	468.3 kB
Polygons	1,723
Materials	10
Uploaded	10/3/07
Last Modified	3/22/14

Daniel
9 models

Estação Pneumatica Festo

Tags
Atuador, Cilindro, clp, componetes, eletro-pneumatica, estação..., festo, furadeira, máquina, plc, Pneumatic,

vonGAL Pallet Conveyor Components

SketchUp 8 Model (960.6 kB)
Google Earth KMZ (1 MB)

Downloads	1,470
Likes	1
.skp File Size	960.6 kB
Polygons	72,268
Materials	10
Uploaded	5/4/11
Last Modified	3/20/14

anonymous
5 models

Various pallet conveyor components manufacured by vonGAL Corporation.

For more information, visit:
http://www.vongal.com

Tags
AS/RS, ASRS, automated handling, automatic conveyor, conveyor, pallet, pallet conveyor, pallet handling, pallet highway, pallet transfer, palletizer, transfer, vonGAL

Teamster factory-in-a-box robotic cell for small sealing application

Downloads	3,192
Likes	0
.skp File Size	5.8 MB
Polygons	33,724
Materials	47
Uploaded	1/12/09
Last Modified	3/20/14

Thomas L.
11 models

Typical cell for making sealings on small boxes with manual loading. Easy to move this cell from place to place, factory to factory.

For more information, visit:
http://www.teamster.se/e/bt/t2x.html

Tags
Teamster robot automation industrial production equipment in-a-box

Parker Gantry Robot GR03C

Parker Gantry Robot Servomotor HLE HPLA

Fig. 5-23: Examples from 3D Warehouse (Trimble, 2014)

The import of COLLADA 1.4 without kinematics and COLLADA 1.5 models including kinematics into a simulation tool is demonstrated for ABB RobotStudio in (Kuhlenkötter et al., 2010), but this is to date not part of the purchasable version which supports only COLLADA 1.4.1 without kinematics (ABB, 2016). This COLLADA import functionality is only part of a few other simulation tools today e.g. Xcelgo Experior, Demo3D/Emulate3D and TaraVR (Dzinic & Yao, 2013, Bockstette, 2013), but generally without kinematics support.

5.5.5 Proposal for mechatronic component models in AutomationML

The novel proposal for exchangeable mechatronic component models should be realisable, based on AutomationML, using CAEX referencing COLLADA and PLCopen XML files. Such an AML component model, as exemplified for the transport car in figure 5-24, could be provided by the manufacturers in the future, e.g. in form of an AML SystemUnitClassLib (cf. subsection 2.9). Such an approach is currently not in the research focus of the AutomationML e.V., as far as is known by the author, probably because the progress is driven by companies from automotive industry which are using concepts with specific workflows based on company internal mechatronic resource libraries.

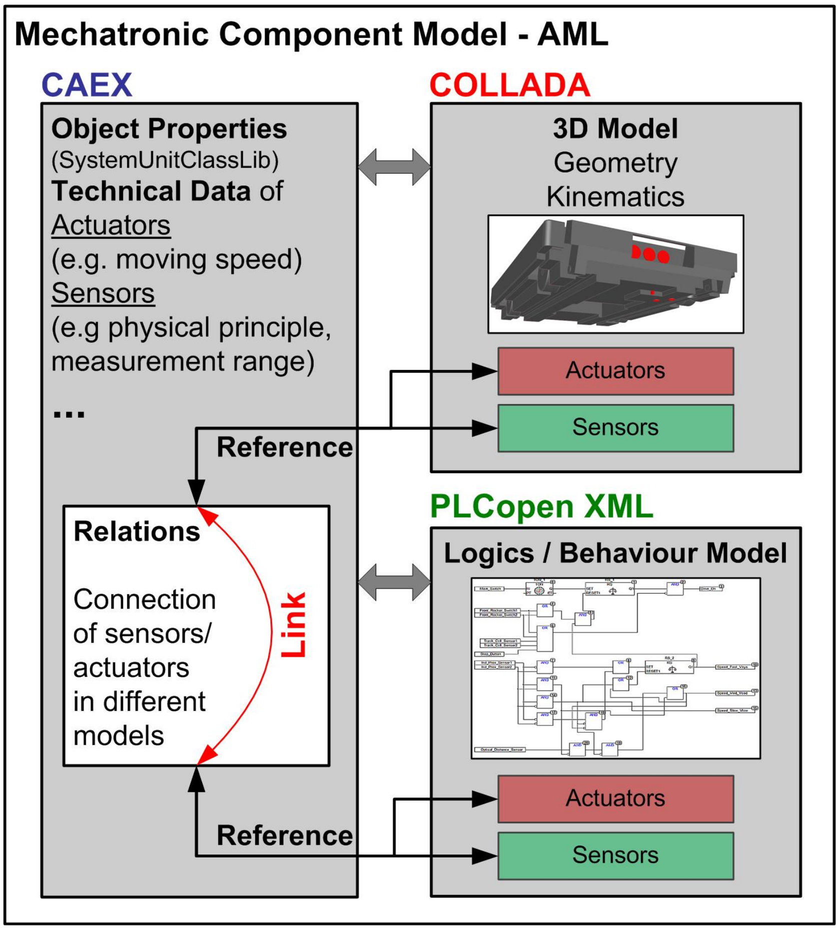


Fig. 5-24: Mechatronic component model in AML

Such AML component models, incorporating XML data files (CAEX, PLCopen XML and COLLADA or perhaps STEP AP242 XML in the future) would allow, after development for importers for simulation tools, the simplified generation of mechatronic component models for the internal libraries of simulation tools. The 3D model should contain geometry and kinematics with sensors and actuators linked to sensor inputs and actuator outputs of the logics/behaviour model and all necessary technical data for parameterisation.

The data interfaces to geometry and kinematics in referenced Collada files, which are also included in current approaches such as those described above. This applies also to the single referencing of PLCopen XML files containing the logics/behaviour, but the approach to couple sensors/actuators in the 3D model via internal links in CAEX, to the same sensors/actuators in the logics/behaviour model, is new. The detailed elaboration and validation of this approach would go beyond the scope of this thesis, but might be interesting for subsequent research.

6 Implementation

According to procedural step 3 of the proposed methodology for a VC simulation study (Fig. 3-1) it is necessary to specify the subsystems, stations and components of the RFAS to be included in the simulation model for VC. As justified above the transportation system has been chosen to start with VC. Hence, the implementation of VC for this thesis starts with the simulation model building of the transportation system.

6.1 Simulation model building of transportation system

The simulation model building comprises the mechatronic component building with CAD preparation and Low-level Component Modelling repeated for every needed component and the one-time High-level Plant Modelling of the transportation system (cf. Fig. 5-8).

6.1.1 Mechatronic component model building

Following the proposed procedure for the systematic model building of mechatronic components (Fig. 5-9) the first step is the component definition in which the transportation system is decomposed into subsystems, stations and components (cf. Fig. 5-6), followed by a specification which of them need to be included in the simulation model.

6.1.1.1 Component Definition

The decomposition of the real transportation system has been described in subsection 4.3. Passive track components (straight-line tracks, curves, signal bars and code bars) as well as active track components (turnouts, stopper units, lifting-positioning unit, data R/W stations, inductive proximity track sensors) have been selected to be part of the simulation model. The following figure 6-1 shows the transportation system and indicates its decomposition, at first to CAD components that will turn into mechatronic components for the simulation model library during subsequent Low-level Component Modelling. None of these components has been in the library of CIROS before starting the modelling.

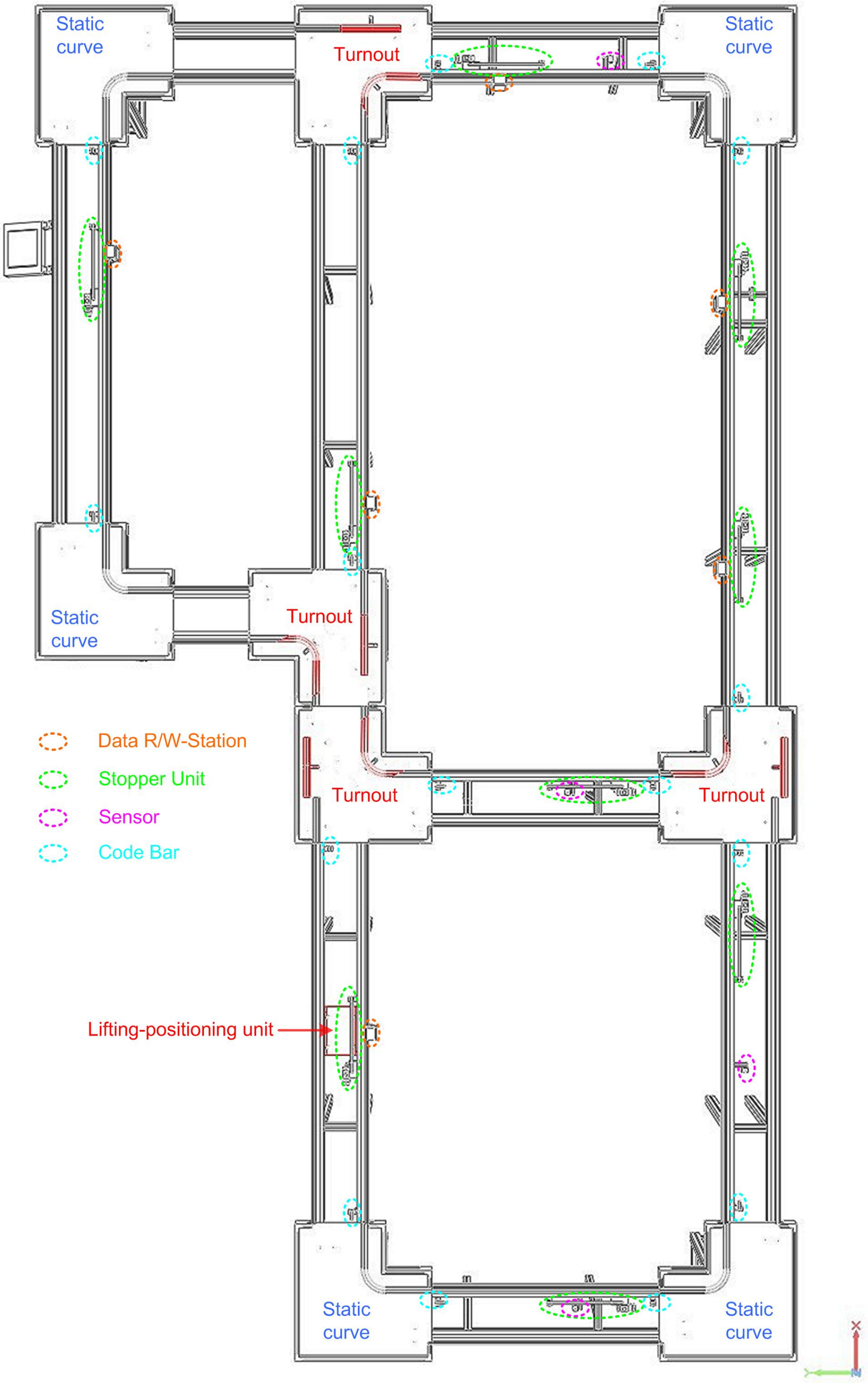


Fig. 6-1: Transportation system assembled from CAD library components (STEP format).

Besides these track components, the transport car with mechanics as well as actuators, sensors, data storage and its behaviour logics has to get an equivalent mechatronic component model.

6.1.1.2 Track Components: Conceptualisation, Data Collection and CAD preparation

The CAD data of the transportation system produced in the forefront of this project turned out to be partly inappropriate for the use with the 3D plant simulation system CIROS. Many parts contained too much detail, other parts were missing, inappropriate assemblies were present and the dimensioning was partly incorrect etc. Hence, it was decided to make new measurements of the hardware to correct false dimensioning of 3D-CAD models and to re-design the 3D-CAD models of the transportation system with subsystems and components appropriate to later simulation model library objects. This 3D-CAD re-design based on the methodology presented in subsection 5.3.4 with its main tasks modularisation, simplification and component model structuring.

The available detailed CAD data of aluminium profiles, used for many constructions within the RFAS (e.g. tracks of the transportation system) must definitely not be used directly for simulation because its complexity will result in an unnecessary high calculation load for the simulation system (Figures 5-11/5-12). Either a simplification or a redesign becomes necessary because no adequately simplified data is available. Principally it would have been possible to simplify these profiles to very simple designs (e.g. outline with only one rectangle) but it has been decided to find a compromise between calculation load and a realistic look of the system inside CIROS.

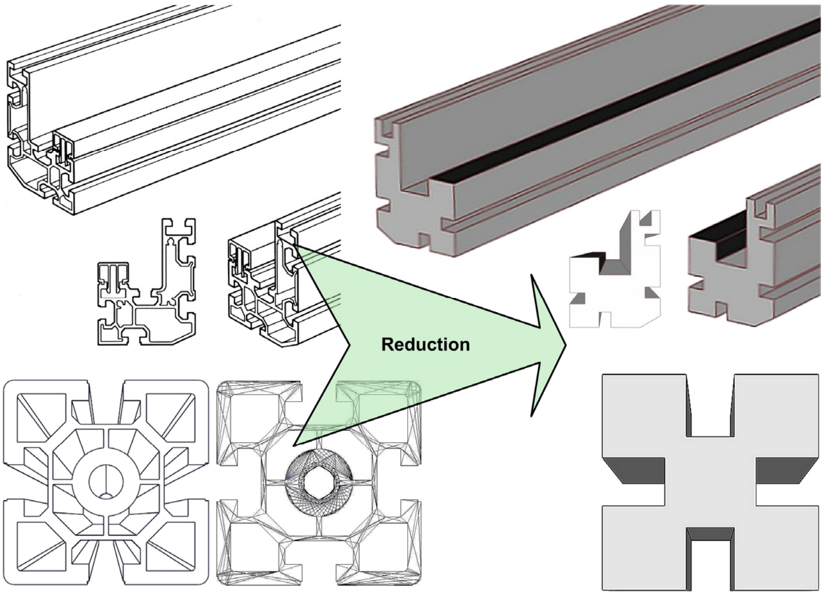


Fig. 6-2: Simplified CAD models of aluminium profiles for the RFAS

The reproduction of original aluminium profile (Fig. 6-2, top left) from (BOSCH, 1992b) is by courtesy of Bosch Rexroth AG. A realistic look of virtual manufacturing systems during 3D simulation seems to be important for the credibility of VC (at least to convince non-specialists in simulation e.g. decision makers from management).

CAD models of passive track components (only static mechanics)

The simple CAD models in figure 6-2 are used for the passive track components namely straight-line tracks, 90° curve (both with base frames) pictured below as well as the code bars.

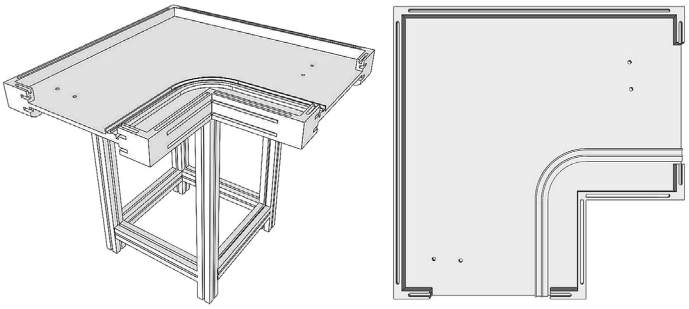


Fig. 6-3: CAD components - Static curve

The CAD components of static curves at the RFAS (Fig. 6-3) do contain neither any sensors (only four holes as possible sensor positions) nor any moving parts. Thus, it is recommended to merge the CAD assembly of the curve to a single part, which is possible by some means or other in the used CAD tools, e.g. using "Shrinkwrap" in Pro/E respectively Creo Parametric (cf. subsection 5.3.4.6).

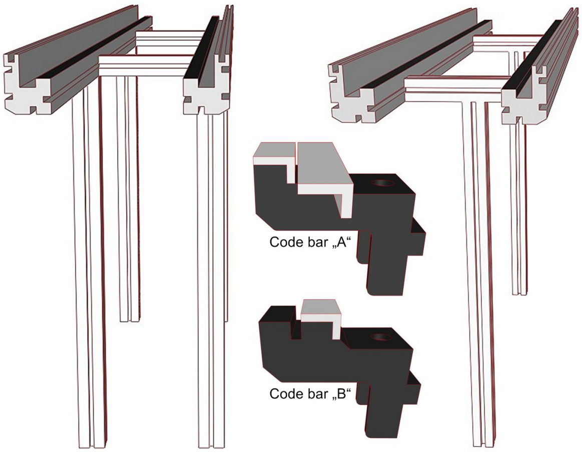


Fig. 6-4: CAD components – Straight-line tracks and code bars

The straight-line tracks (Fig. 6-4) are also stored with separate base frames in the library. The signal bar (cf. Fig. 4-18) has not been defined as single CAD component like the code bars because it is only used inside the stopper unit (cf. Fig. 6-5) at the RFAS. For a commercial version and convenient use of such a library, all components should be available as single component and as typical unit, e.g. sensor, stopper block and signal bar etc. and all together as stopper unit.

CAD models of active track components

The active track components can be divided in electro-pneumatic components with moving actuators and sensors, and electric components with only sensors/communication systems.

Electro-pneumatic components

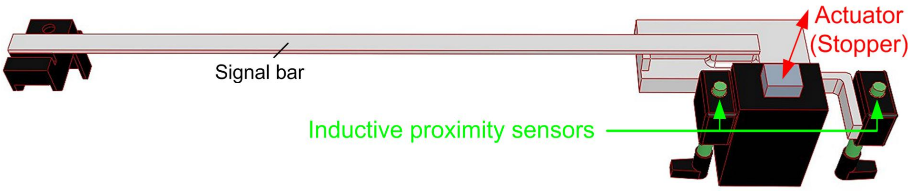


Fig. 6-5: CAD components – Stopper unit

The crucial factor for best usability of this stopper unit (Fig. 6-5) for simulation with CIROS is the separation of the moveable stopper and both proximity sensors (cf. subsection 5.3.4.6). The other static parts of the CAD model can be merged to one part.

The originally designed CAD model of the stopper unit holds many details such as holes and roundings that are irrelevant for simulation and would increase the graphics calculation load. If using Creo Parametric’s feature Shrinkwrap simplification and merging to a part (excluding stopper and sensors) can be done at once, other CAD tools require two steps for simplification and finally storing of assemblies as merged part. The following figures 6-6 – 6-8 exemplify the simplification of one element of the stopper unit conducted in Solid Edge.

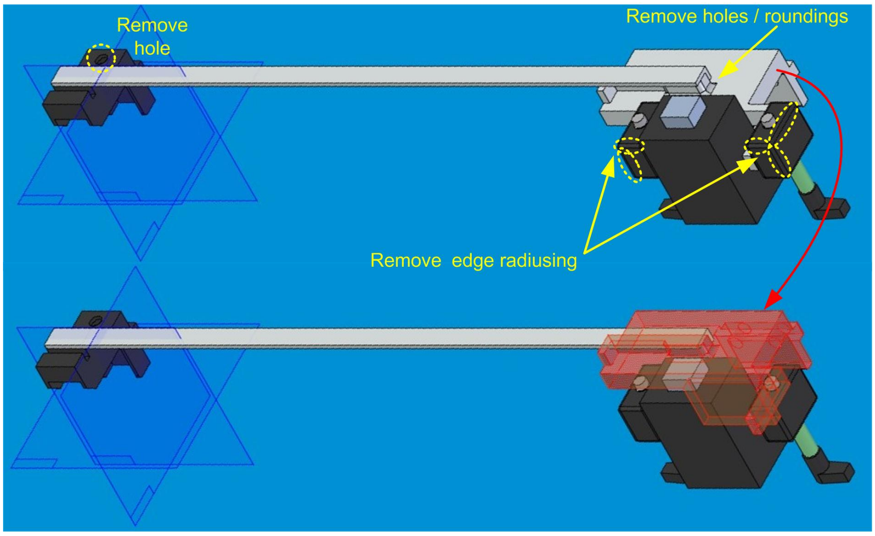


Fig. 6-6: Example for CAD simplification – Stopper unit

First holes and roundings at the base part for the signal bar have been simplified. For this, the part to be simplified has been cut free (Fig. 6-7).

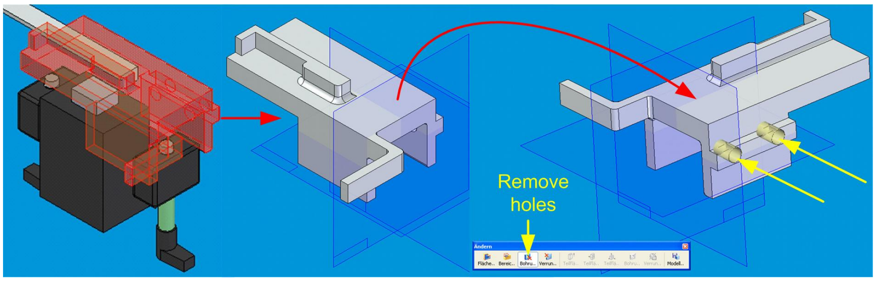


Fig. 6-7: Example for CAD simplification – Removing holes

Solid Edge, as other CAD tools, allows the selection of holes individually and it is possible to use filter function with adjustable diameter to remove holes automatically. Roundings have been removed in a similar way afterwards (Fig. 6-8). If it is intended to have roundings in the model, it is recommended to design only few facets (cf. CAD guidelines subsection 5.3.4.7). Current versions of Solid Edge provide the feature to automatically simplify an entire assembly.

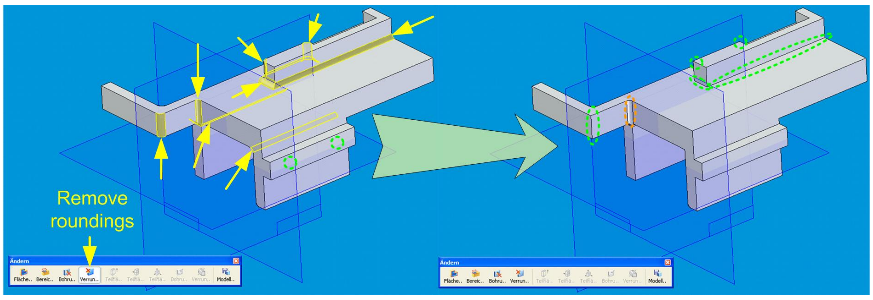


Fig. 6-8: Example for CAD simplification – Removing roundings

The same procedure has been repeated for the other parts of the stopper unit (cf. fig. 6-6) resulting in the simplified version shown in figure 6-9.

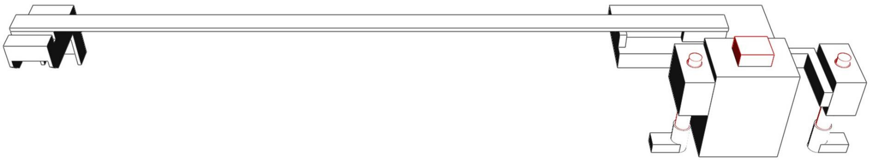


Fig. 6-9: CAD components – Stopper unit after simplification

The final version has been stored in the CAD library, and after Low-level Component Modelling, it is used nine times in the entire model of the transportation system.

In contrast to the curves, the turnouts are not completely static. They hold two cylinders moving guidance profiles and inductive proximity sensors. It is essential for later low-level modelling to keep the moving parts and the sensors separated from the remaining static parts in the CAD model. To relieve low-level modelling the guidance profiles and the pistons of cylinders have been merged to one moveable part (Fig. 6-10). Details unnecessary for intended simulation such as the 5/2 directional control valve, electrical and pneumatical wiring etc. (cf. Fig. 4-23) have not been modelled at all.

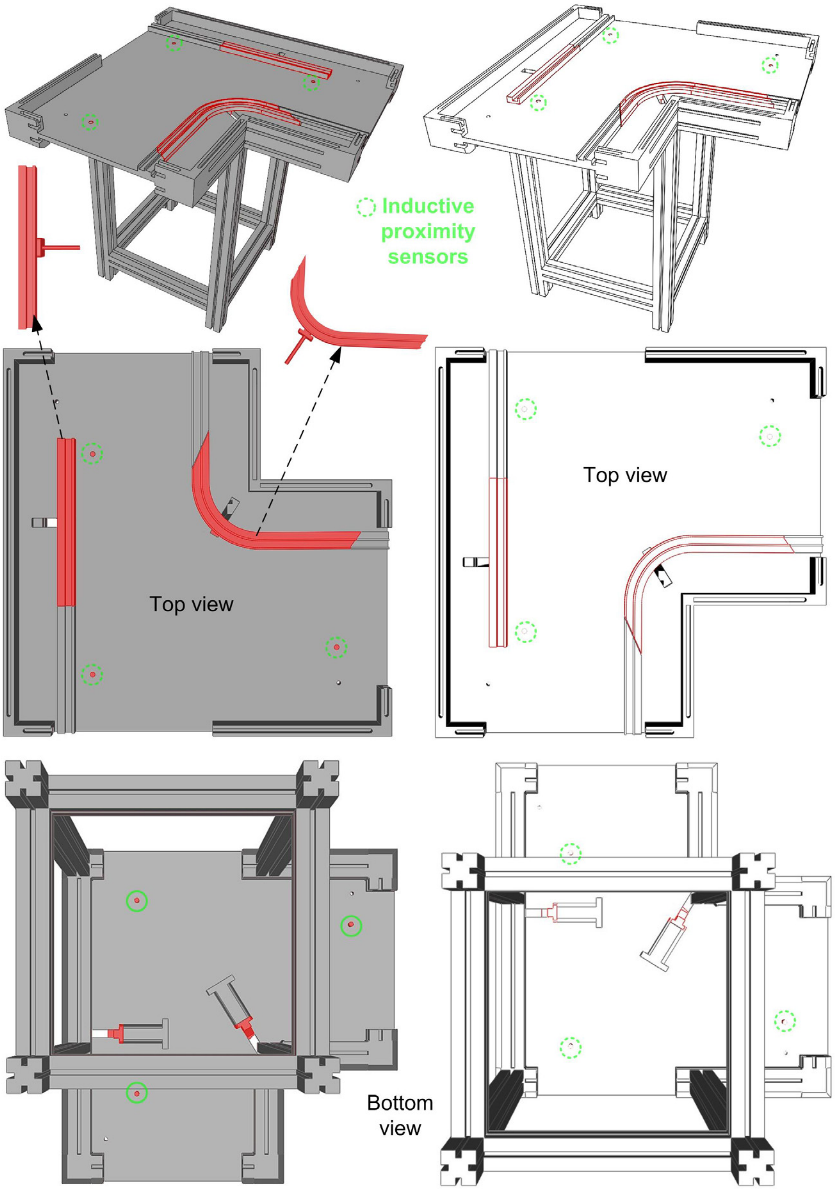


Fig. 6-10: CAD components – Turnouts

Thus, when merging the turnout to a part using Shrinkwrap, one has to consider that all parts are selected by default, but it is necessary to exclude the moving guidance profiles and the sensors from the merging. Moveable parts and sensors are indicated with red colour in the different views in figure 6-10.

The Lifting-positioning unit mounted inside SCARA robot cell has been simplified keeping only the fixed grippers and the locating pins on the separate moveable part indicated by red colour (Fig. 6-11).

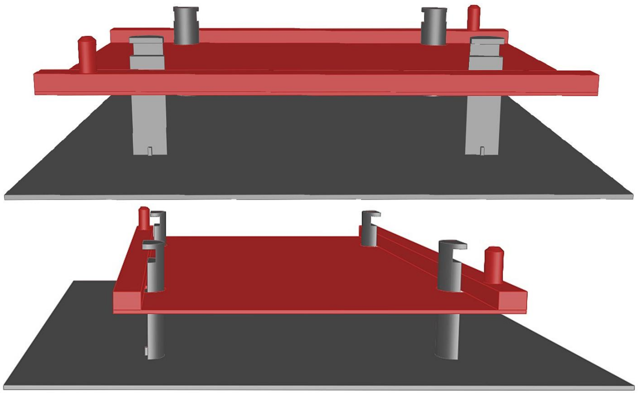


Fig. 6-11: CAD components – Lifting-positioning unit

There are no more components with moving part in the transportation system at the RFAS, but some electrical components to play a role in simulation.

Electric components

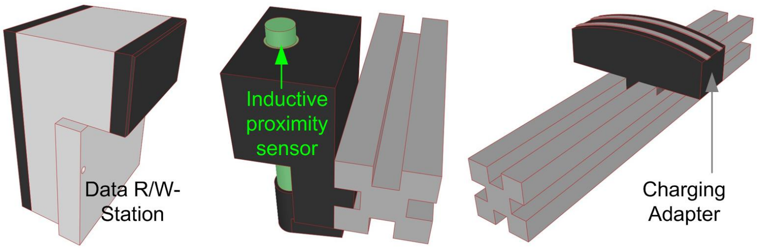


Fig. 6-12: CAD components – Data R/W-Station, Track Sensor and Charging Adapter

The CAD model of the Data R/W-Station is intended to be equipped with the simulated transceiver function in CIROS. The inductive proximity sensor in the track sensor (Fig. 6-12) has to be separated again. The charging adapter has been designed and stored in the CAD library for possible later simulation of battery charging.

6.1.1.3 Transport car: Conceptualisation, Data Collection and CAD preparation

Based on the model conceptualisation of the entire transportation system (cf. Fig. 5-10) a more detailed conceptual model of the transport car has been developed (Fig. 6-13). The underlying investigation how the real transport car works has been presented in subsection 4.3.1. Likewise, the available and collected technical data are presented in this subsection.

This component model conceptualisation for the simulation model of the transport car can be divided into conceptualisation and CAD design of the mechanical 3D model (Fig. 6-14) and the conceptualisation of the logics/behaviour (Fig. 6-15).

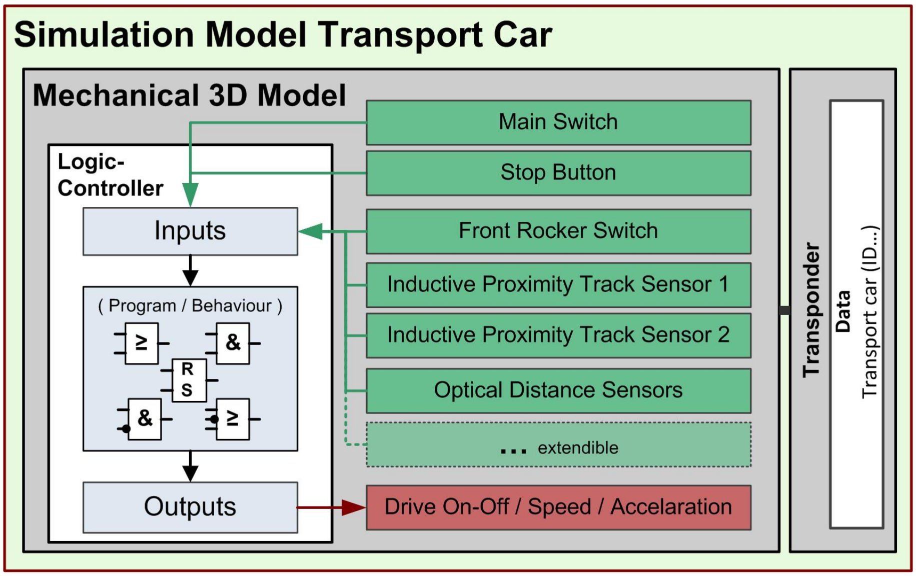


Fig. 6-13: Simulation model conceptualisation of transport car

This has turned out with several simplifications compared to the real TW40. It has been forbeared to design a moveable front rocker (activating switches behind) for the first mechanical 3D model to start, because this would have required a complex kinematical modelling in CIROS (kinematics cannot be transferred from CAD). The functionality of the real front rocker switch will be substituted by exterior sensors in CIROS that can be activated without movement of the front rocker. The moveable stop button (cf. Fig. 4-12) has been designed as single static CAD object, its function will be performed by a sensor in CIROS. Besides the stop button, the optical distance sensors and the inductive proximity sensors have been designed as separate CAD objects too (recognisable by red colour in figure 6-14).

Several details such as the service connector and the fuse have not been designed because they are not necessary for simulation. After weighing up modelling effort in CIROS and additional benefit, the seven-segment display has been omitted for the start. If it turns out during simulation that a status display for the transport car is useful, a retrofit of the model will be done. The guidance roller and the cornering wheel have been left out, because they are not necessary when using the Transport Simulation in CIROS. The major simplification results from omitting to model the battery charging/discharging and sleeping modes of the transport car. Due to the comparatively small track layout and two charging units of the real system, this has been regarded as to be dispensable for the VC of the RFAS, but it would be a challenging modelling task for the future work.

An exemplary use-case specific mounting plate with three acrylic cylinder cylinders is shown in figure 6-14. Naturally, the mounting plate and the acrylic cylinders have to be separate parts for flexible use of the CAD model.

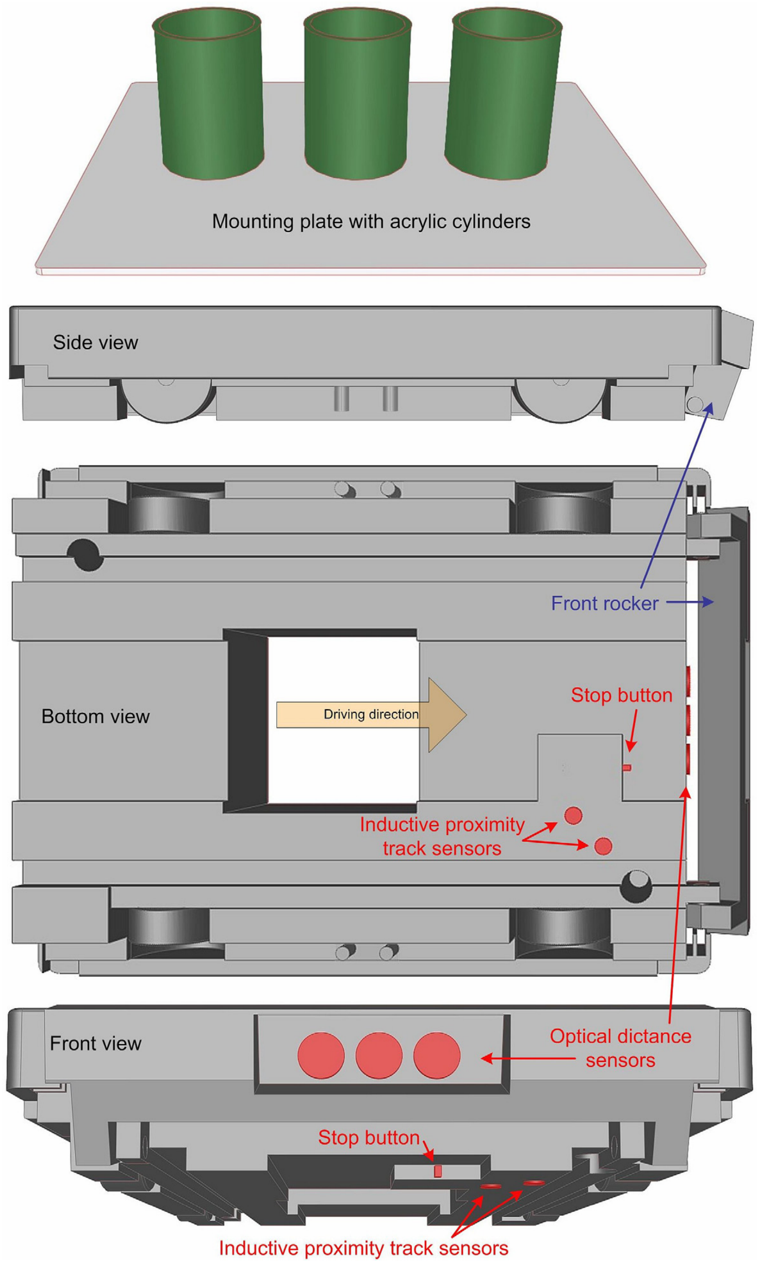


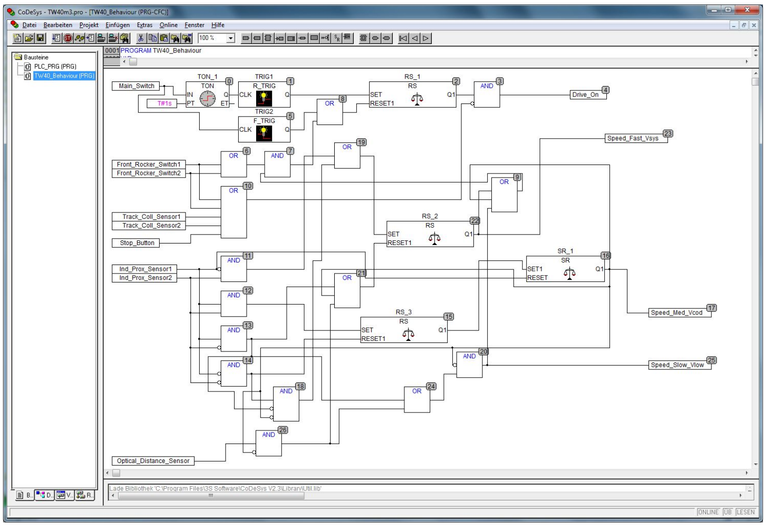
Fig. 6-14: CAD components – Transport car TW40

Finally, all track components and the transport car have been exported from CAD tools as STEP format for later import in CIROS. Solid Edge and Creo Parametric both provide adjustable STEP export filter with many possible options that has been tested among each other and with import in CIROS.

Expectedly, it has not been possible to transfer any movement relations, limited degrees of freedom and limited translatory motion of parts etc., not to mention more complex kinematics, between used CAD tools or to CIROS. Even the transfer of all other information included in the CAD models such as part naming, material properties or colour is not assured, e.g. it was not possible to transfer material data with any of the tested options.

Conceptual model of transport car logics/behaviour

Because of the complex behaviour of this component, a special conceptual model has been developed for the transport car. A Function Block Diagram (FBD) has been designed in CoDeSys 2.3 (termed CFC - Continuous Function Chart), based on the textual description (supported with figures) provided in the vendor's manual. The detailed description of the transport car's behaviour, gathered from this manual, is given in subsection 4.3.1. Even though CIROS currently does not support the import of logics/behaviour descriptions, such a FBD is useful for the manual programming of the logics controller included in the simulation model of the transport car.



```
0001 PROGRAM TW40_Behaviour
```

```
Lade Bibliothek 'C:\Program Files\3S Software\CoDeSys V2.3\Library\Util.lib'
```

Fig. 6-15: Conceptual Model – Function Block Diagram (CFC) transport car

The Function Block Diagram in figure 6-15 does not completely reproduce the functionality and behaviour of the real transport car. During component model conceptualisation starting with 3D (Fig. 6-14), it has been decided to relinquish on the simulation of the seven-segment display as well as transport car sleep modes and as mentioned no battery simulation and sleep modes have been implemented. Because the currently used version of CIROS does not support physics simulation the 2g shock sensor has been initially left out too.

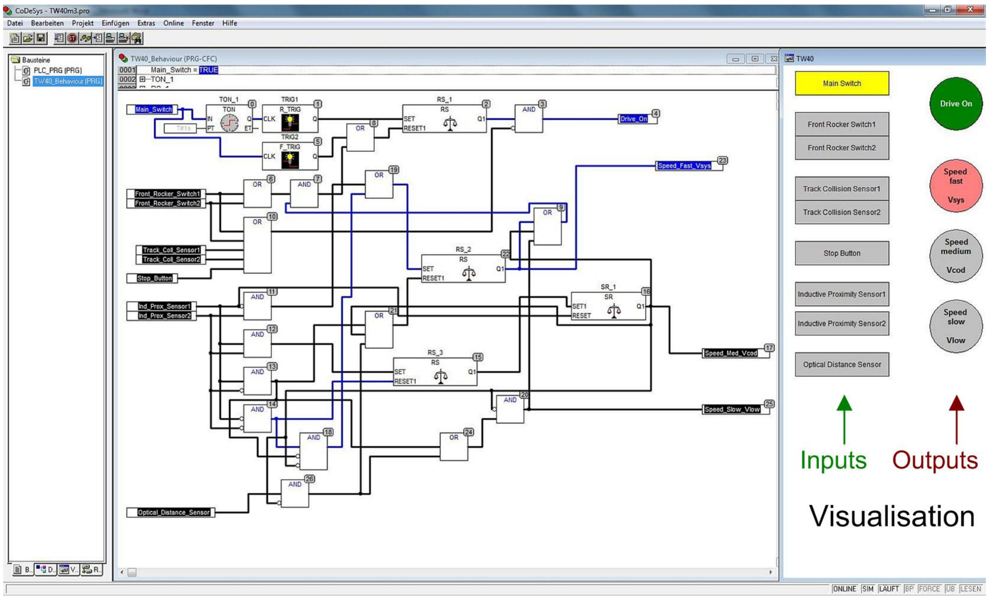


Fig. 6-16: V&V of conceptual model transport car

A visualisation build in CoDeSys allows a convenient testing of the conceptual model by activating the sensor inputs via buttons and observing outputs (Fig. 6-16).

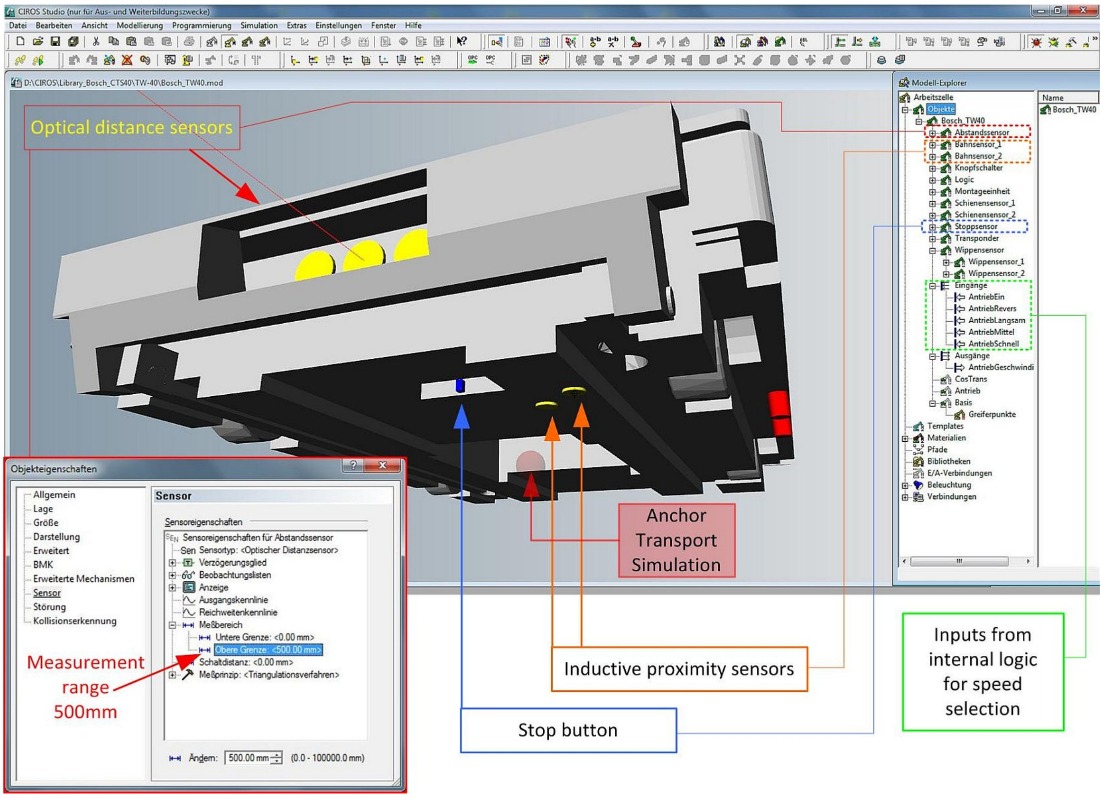
Such a conceptual model represents a logics/behaviour model that could be likewise part of a mechatronic component model as proposed in subsection 5.4.5. For this purpose, it has to be generated with an IDE supporting PLCopen XML, such as MultiProg (KW-Software). The approach to use e.g. FBD for future exchangeable logics/behaviour models has been discussed, with the transport car as use-case, on an AutomationML workshop with positive feedback (Hoffmann, 2013).

6.1.1.4 Low-level Component Modelling in CIROS

The Low-level Component Modelling has been conducted according to the proposed systematic procedure presented in 5.3.4.8. After import of CAD models in STEP format, this task comprises geometrical modelling, functional modelling and electrical modelling. Examples are given below.

Component model - Transport car

This component model is based on the CAD model in figure 6-14 and has been equipped with sensors, logic controller and a transponder according to figure 6-13. Currently, the transponder stores only an ID number (1-4) of the transport car and is read by the data R/W station (Fig. 6-23).



Sensor

Sensoreigenschaften

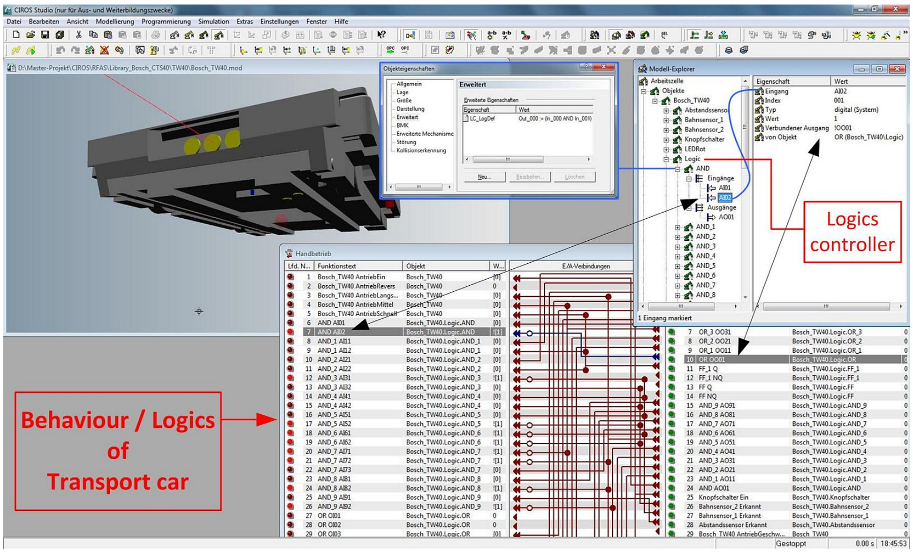
- Sensoreigenschaften für Abstandssensor
  - Sensortyp: <Optischer Distanzsensor>
  - Verzögerungsglied
  - Beobachtungslisten
  - Anzeige
  - Ausgangskennlinie
  - Reichweitenkennlinie
  - Meßbereich
    - Untere Grenze: <0.00 mm>
    - Obere Grenze: <500.00 mm>
  - Schaltdistanz: <0.00 mm>
  - Meßprinzip: <Triangulationsverfahren>

Ändern: 500.00 mm (0.0 - 100000.0 mm)

Fig. 6-17: Low-level Component Modelling –Transport car

After assignment of sensor functionality to the correspondent parts in the 3D model, these sensors have to be parameterised. An example is given for the optical distance sensor; the measurement range has been set to 500 mm inside the sensor property window (red rectangle). The anchor for the Transport Simulation (cf. subsection 5.2.1) is used to couple the transport car to the segments and to guide it along the tracks (e.g. Fig. 6-19). The inputs (green rectangle) are used to read the outputs from logics/behaviour model determined inside the logics controller (Fig. 6-18).

CIROS, as it is usually the case for simulation tools for VC, does not allow the import of logic models, thus logics from a conceptual model (Fig. 6-15) has to be implemented manually. An extract is shown in figure 6-18. This logic determines the speed of the transport car (slow, medium, fast) and starts or stops the drive depending on the inputs from switches/sensors. The internal logic of the transport car is connected to the Transport Simulation, and allows e.g. a stopped transport car and a moving transport car on the same track, or transport cars with different speeds on the same track.



Eigenschaft	Wert
Eingang	AI02
Index	001
Typ	digital (System)
Wert	1
Verbundener Ausgang	!OO01
von Objekt	OR (Bosch_TW40\Logic)

Lfd. N...	Funktionstext	Objekt	W...
1	Bosch_TW40 AntriebEin	Bosch_TW40	[0]
2	Bosch_TW40 AntriebRevers	Bosch_TW40	0
3	Bosch_TW40 AntriebLangs...	Bosch_TW40	[0]
4	Bosch_TW40 AntriebMittel	Bosch_TW40	[0]
5	Bosch_TW40 AntriebSchnell	Bosch_TW40	[0]
6	AND AI01	Bosch_TW40.Logic.AND	[0]
7	AND AI02	Bosch_TW40.Logic.AND	![1]
8	AND_1 AI11	Bosch_TW40.Logic.AND_1	[0]
9	AND_1 AI12	Bosch_TW40.Logic.AND_1	[0]
10	AND_2 AI21	Bosch_TW40.Logic.AND_2	[0]
11	AND_2 AI22	Bosch_TW40.Logic.AND_2	[0]
12	AND_3 AI31	Bosch_TW40.Logic.AND_3	![1]
13	AND_3 AI32	Bosch_TW40.Logic.AND_3	[0]
14	AND_4 AI41	Bosch_TW40.Logic.AND_4	[0]
15	AND_4 AI42	Bosch_TW40.Logic.AND_4	[0]
16	AND_5 AI51	Bosch_TW40.Logic.AND_5	[0]
17	AND_5 AI52	Bosch_TW40.Logic.AND_5	![1]
18	AND_6 AI61	Bosch_TW40.Logic.AND_6	![1]
19	AND_6 AI62	Bosch_TW40.Logic.AND_6	![1]
20	AND_7 AI71	Bosch_TW40.Logic.AND_7	![1]
21	AND_7 AI72	Bosch_TW40.Logic.AND_7	![1]
22	AND_7 AI73	Bosch_TW40.Logic.AND_7	[0]
23	AND_8 AI81	Bosch_TW40.Logic.AND_8	[0]
24	AND_8 AI82	Bosch_TW40.Logic.AND_8	![1]
25	AND_9 AI91	Bosch_TW40.Logic.AND_9	[0]
26	AND_9 AI92	Bosch_TW40.Logic.AND_9	![1]
27	OR OI01	Bosch_TW40.Logic.OR	0
28	OR OI02	Bosch_TW40.Logic.OR	0
29	OR OI03	Bosch_TW40.Logic.OR	[0]

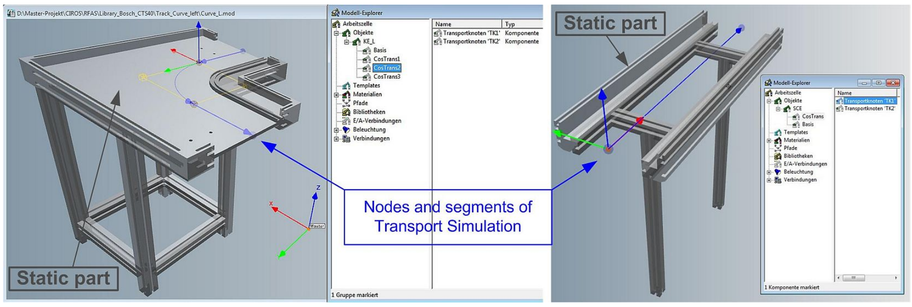
7	OR_3 OO31	Bosch_TW40.Logic.OR_3	0
8	OR_2 OO21	Bosch_TW40.Logic.OR_2	0
9	OR_1 OO11	Bosch_TW40.Logic.OR_1	0
10	OR OO01	Bosch_TW40.Logic.OR	0
11	FF_1 Q	Bosch_TW40.Logic.FF_1	0
12	FF_1 NQ	Bosch_TW40.Logic.FF_1	0
13	FF Q	Bosch_TW40.Logic.FF	0
14	FF NQ	Bosch_TW40.Logic.FF	0
15	AND_9 AO91	Bosch_TW40.Logic.AND_9	0
16	AND_8 AO81	Bosch_TW40.Logic.AND_8	0
17	AND_7 AO71	Bosch_TW40.Logic.AND_7	0
18	AND_6 AO61	Bosch_TW40.Logic.AND_6	0
19	AND_5 AO51	Bosch_TW40.Logic.AND_5	0
20	AND_4 AO41	Bosch_TW40.Logic.AND_4	0
21	AND_3 AO31	Bosch_TW40.Logic.AND_3	0
22	AND_2 AO21	Bosch_TW40.Logic.AND_2	0
23	AND_1 AO11	Bosch_TW40.Logic.AND_1	0
24	AND AO01	Bosch_TW40.Logic.AND	0
25	Knopfschalter Ein	Bosch_TW40.Knopfschalter	0
26	Bahnsensor_2 Erkannt	Bosch_TW40.Bahnsensor_2	0
27	Bahnsensor_1 Erkannt	Bosch_TW40.Bahnsensor_1	0
28	Abstandssensor Erkannt	Bosch_TW40.Abstandssensor	0
29	Bosch_TW40 AntriebGeschw...	Bosch_TW40	0

Fig. 6-18: Low-level Component Modelling –Logics of transport car

This is a novel application of CIROS and its Transport Simulation to a track-bound transportation system with self-driving ('intelligent') transport cars on passive tracks. Previously it has been only applied to transport systems with active tracks (e.g. belt drive). Thus, when the concept for the use of a logic controller inside the transport car that is interacting with the Transport Simulation in the tracks has been developed not all elements necessary for the CTS system had been in the mechanisms library. Thanks to a developer from RIF, this problem has been solved quickly, and since then it is all part of the CIROS standard library. Hence, the first part of the hypothesis in subsection 5.2.2 has been corroborated; it is possible to model the Bosch CTS with its self-driving transport cars on passive tracks with CIROS Studio.

Component models – Curve and Tracks

The modelling of the static track components (curve and straight-line tracks in Fig. 6-19) is comparatively simple, only segments and nodes of the Transport Simulation have to be added.



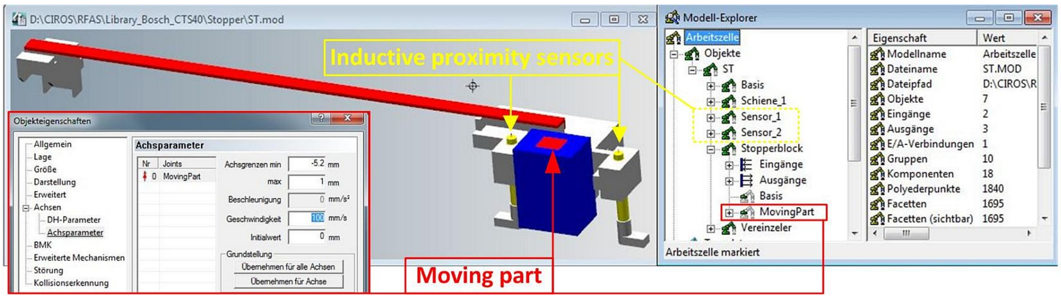
Name	Typ
Transportknoten 'TK1'	Komponente
Transportknoten 'TK2'	Komponente

Fig. 6-19: Component models – Curve and straight-line tracks

The positioning of these elements has to be accurate, thereby the transport cars move geometrically correct on the tracks; this means the track idlers (cf. Fig. 4-11) of the transport cars move correctly inside the guidance profiles (cf. Fig. 7-2) and the transport car sensors are correctly activated by the code bars/signal bars inside the tracks (cf. Fig. 7-3). The nodes provide a snapping functionality for easier High-level Plant Modelling of transportation systems.

Component model - Stopper unit

Due to the appropriate CAD preparation (separation of static parts, moving parts and sensors according to subsection 5.3.4.6), the component modelling is greatly simplified.


D:\CIROS\RFAS\Library_Bosch_CTS40\Stopper\ST.mod

Objekteigenschaften

Achsparameter

Nr	Joints
0	MovingPart

Achsgrenzen min: -5.2 mm
max: 1 mm
Beschleunigung: 0 mm/s²
Geschwindigkeit: 100 mm/s
Initialwert: 0 mm

Modell-Explorer

Eigenschaft	Wert
Modellname	Arbeitszelle
Dateiname	ST.MOD
Dateipfad	D:\CIROS\R
Objekte	7
Eingänge	2
Ausgänge	3
E/A-Verbindungen	1
Gruppen	10
Komponenten	18
Polyederpunkte	1840
Facetten	1695
Facetten (sichtbar)	1695

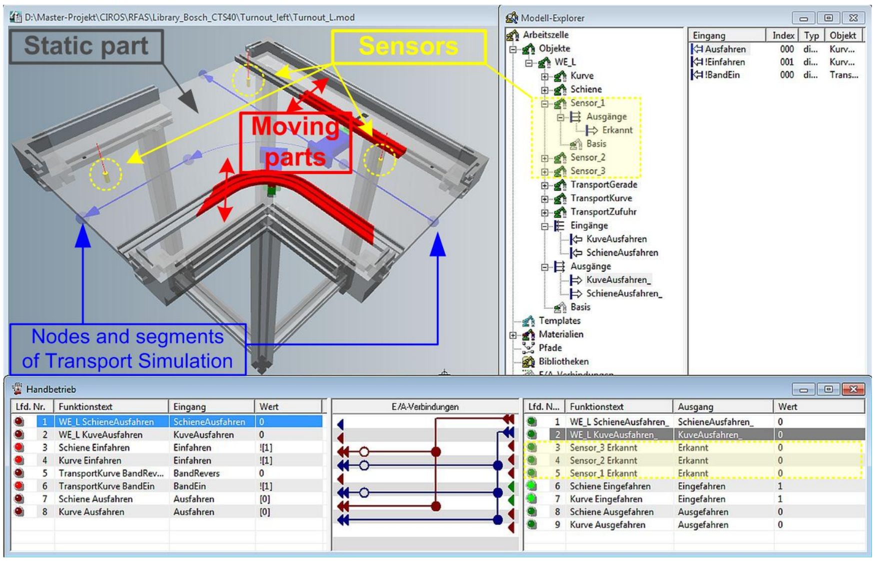
Arbeitszelle markiert

Fig. 6-20: Component model – Stopper unit

The component model (Fig. 6-20) contains three sensors (two proximity sensors, one end position sensor) and the stopper block as moving part. Stroke and speed of this moving part have to be parameterised (object property window – red rectangle). This mechanism (one-way push cylinder) provides an end position sensor by default. The proximity sensors have to be parameterised as well.

Component model - Turnout

The component models for the turnouts are a bit more complex. The CAD modelling of the two mechanically different turnouts has been also conducted according to the recommended structuring (cf. Fig. 6-10). The importance of this appropriate structuring of CAD models can be estimated by comparison with the import of unsuitable structured CAD models depicted in figure 5-16. Due to possibly reversed driving direction of transport cars passing the turnout, this results in four turnout configurations, which have been stored in the library.



Eingang	Index	Typ	Objekt
Ausfahren	000	di...	Kurv...
!Einfahren	001	di...	Kurv...
!BandEin	000	di...	Trans...

Lfd. Nr.	Funktionstext	Eingang	Wert
1	WE_L SchieneAusfahren	SchieneAusfahren	0
2	WE_L KuveAusfahren	KuveAusfahren	0
3	Schiene Einfahren	Einfahren	![1]
4	Kurve Einfahren	Einfahren	![1]
5	TransportKurve BandRev...	BandRevers	0
6	TransportKurve BandEin	BandEin	![1]
7	Schiene Ausfahren	Ausfahren	[0]
8	Kurve Ausfahren	Ausfahren	[0]

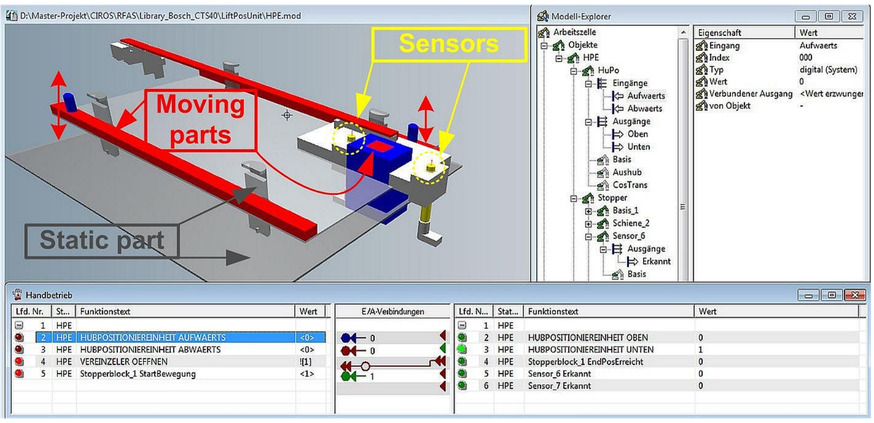
Lfd. N...	Funktionstext	Ausgang	Wert
1	WE_L SchieneAusfahren_	SchieneAusfahren_	0
2	WE_L KuveAusfahren_	KuveAusfahren_	0
3	Sensor_3 Erkannt	Erkannt	0
4	Sensor_2 Erkannt	Erkannt	0
5	Sensor_1 Erkannt	Erkannt	0
6	Schiene Eingefahren	Eingefahren	1
7	Kurve Eingefahren	Eingefahren	1
8	Schiene Ausgefahren	Ausgefahren	0
9	Kurve Ausgefahren	Ausgefahren	0

Fig. 6-21: Component model – Turnout

This turnout component model (Fig. 6-21) contains two moving parts (red guidance profiles), with two-way push cylinder mechanism, and seven sensors (three proximity sensors and four end position sensors). The switching of the direction of Transport Simulation is connected to the moving parts via the internal logic (bottom of Fig. 6-21). Actuators and sensors have to be parameterised as described above.

Component model – Lifting-positioning unit

The component model of the lifting-positioning unit contains a static part, a moving part with two end position sensors, and three sensors inside the included stopper unit (Fig. 6-22). The blue locating pins (cf. Fig. 4-26) move into the locating bushings of the transport car (cf. Figs. 4-12 and 6-14) when lifting the transport car.



Eigenschaft	Wert
Eingang	Aufwaerts
Index	000
Typ	digital (System)
Wert	0
Verbundener Ausgang	<Wert erzwunger
von Objekt	-

Lfd. Nr.	St...	Funktionstext	Wert
1	HPE		
2	HPE	HUBPOSITIONIEREINHEIT AUFWAERTS	<0>
3	HPE	HUBPOSITIONIEREINHEIT ABWAERTS	<0>
4	HPE	VEREINZELER OEFFNEN	![1]
5	HPE	Stopperblock_1 StartBewegung	<1>

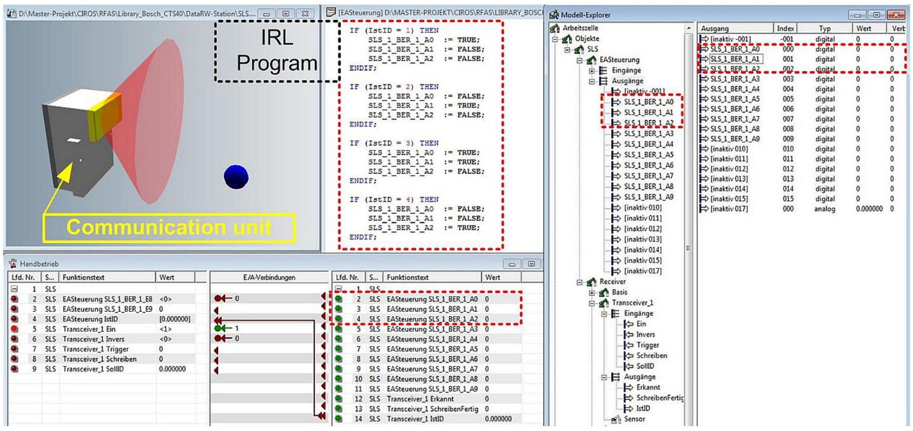
Lfd. N...	Stat...	Funktionstext	Wert
1	HPE		
2	HPE	HUBPOSITIONIEREINHEIT OBEN	0
3	HPE	HUBPOSITIONIEREINHEIT UNTEN	1
4	HPE	Stopperblock_1 EndPosErreicht	0
5	HPE	Sensor_6 Erkannt	0
6	HPE	Sensor_7 Erkannt	0

Fig. 6-22: Component model – Lifting-positioning unit

This lifting has been realized in this case by a moving node of the Transport Simulation, which moves the transport car with its anchor snapped to this node. Generally, such movement is realized by using the gripping functionality of CIROS. Here, as in the examples above, actuators and sensors have to be parameterised too.

Component model – Data R/W station

This component model of a communication unit, the data R/W station, is a static part without sensors or actuators (Fig. 6-23), but it makes use of an IRL program. This unit reads the data stored in the transponder of the transport car, currently only its ID number.



```
IF (IstID = 1) THEN
    SLS_1_BER_1_A0  := TRUE;
    SLS_1_BER_1_A1  := FALSE;
    SLS_1_BER_1_A2  := FALSE;
ENDIF;

IF (IstID = 2) THEN
    SLS_1_BER_1_A0  := FALSE;
    SLS_1_BER_1_A1  := TRUE;
    SLS_1_BER_1_A2  := FALSE;
ENDIF;

IF (IstID = 3) THEN
    SLS_1_BER_1_A0  := TRUE;
    SLS_1_BER_1_A1  := TRUE;
    SLS_1_BER_1_A2  := FALSE;
ENDIF;

IF (IstID = 4) THEN
    SLS_1_BER_1_A0  := FALSE;
    SLS_1_BER_1_A1  := FALSE;
    SLS_1_BER_1_A2  := TRUE;
ENDIF;
```

Lfd. Nr.	S...	Funktionstext	Wert
1	SLS		
2	SLS	EASteuerung SLS_1_BER_1_E8	<0>
3	SLS	EASteuerung SLS_1_BER_1_E9	0
4	SLS	EASteuerung IstID	[0.000000]
5	SLS	Transceiver_1 Ein	<1>
6	SLS	Transceiver_1 Invers	<0>
7	SLS	Transceiver_1 Trigger	0
8	SLS	Transceiver_1 Schreiben	0
9	SLS	Transceiver_1 SollID	0.000000

Lfd. Nr.	S...	Funktionstext	Wert
1	SLS		
2	SLS	EASteuerung SLS_1_BER_1_A0	0
3	SLS	EASteuerung SLS_1_BER_1_A1	0
4	SLS	EASteuerung SLS_1_BER_1_A2	0
5	SLS	EASteuerung SLS_1_BER_1_A3	0
6	SLS	EASteuerung SLS_1_BER_1_A4	0
7	SLS	EASteuerung SLS_1_BER_1_A5	0
8	SLS	EASteuerung SLS_1_BER_1_A6	0
9	SLS	EASteuerung SLS_1_BER_1_A7	0
10	SLS	EASteuerung SLS_1_BER_1_A8	0
11	SLS	EASteuerung SLS_1_BER_1_A9	0
12	SLS	Transceiver_1 Erkannt	0
13	SLS	Transceiver_1 SchreibenFertig	0
14	SLS	Transceiver_1 IstID	0.000000

Ausgang	Index	Typ	Wert	Verb
[inaktiv -001]	-001	digital	0	0
SLS_1_BER_1_A0	000	digital	0	0
SLS_1_BER_1_A1	001	digital	0	0
SLS_1_BER_1_A2	002	digital	0	0
SLS_1_BER_1_A3	003	digital	0	0
SLS_1_BER_1_A4	004	digital	0	0
SLS_1_BER_1_A5	005	digital	0	0
SLS_1_BER_1_A6	006	digital	0	0
SLS_1_BER_1_A7	007	digital	0	0
SLS_1_BER_1_A8	008	digital	0	0
SLS_1_BER_1_A9	009	digital	0	0
[inaktiv 010]	010	digital	0	0
[inaktiv 011]	011	digital	0	0
[inaktiv 012]	012	digital	0	0
[inaktiv 013]	013	digital	0	0
[inaktiv 014]	014	digital	0	0
[inaktiv 015]	015	digital	0	0
[inaktiv 017]	000	analog	0.000000	0

Fig. 6-23: Component model – Data R/W station

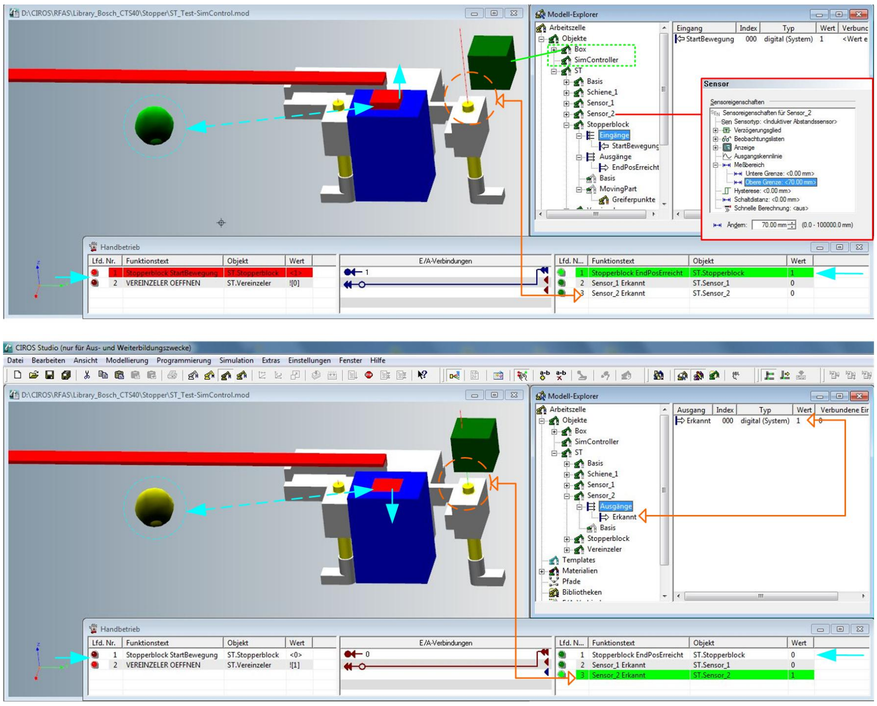
This ID number is transferred to the PLC program. The transport car numbers 1-4 are coded by the IRL program to three bits (Fig. 6-23, red dotted rectangles). Besides two bits for handshaking with PLC all other bits are currently unused by PLC programs.

6.1.1.5 Verification and Validation of component models

The V&V of component models has to ensure that the models have been correctly implemented (verification) and that the models (especially the transport car) adequately represent the behaviour of real components (validation).

Stopper unit

The V&V of the stopper unit is presented here as an example for the V&V of track components. Additionally to the completed component model (Fig. 6-20), a SimController has been implemented to allow manual operation for testing and a box to activate the sensors (Fig. 6-24, green dotted rectangle).



Handbetrieb

Lfd. Nr.	Funktionstext	Objekt	Wert
1	Stopperblock StartBewegung	ST.Stopperblock	<1>
2	VEREINZELER OEFFNEN	ST.Vereinzeler	![0]

Lfd. N...	Funktionstext	Objekt	Wert
1	Stopperblock EndPosErreicht	ST.Stopperblock	1
2	Sensor_1 Erkannt	ST.Sensor_1	0
3	Sensor_2 Erkannt	ST.Sensor_2	0

Sensor: Oberе Grenze: <70.00 mm>; Ändern: 70.00 mm (0.0 - 100000.0 mm)

Handbetrieb

Lfd. Nr.	Funktionstext	Objekt	Wert
1	Stopperblock StartBewegung	ST.Stopperblock	<0>
2	VEREINZELER OEFFNEN	ST.Vereinzeler	![1]

Lfd. N...	Funktionstext	Objekt	Wert
1	Stopperblock EndPosErreicht	ST.Stopperblock	0
2	Sensor_1 Erkannt	ST.Sensor_1	0
3	Sensor_2 Erkannt	ST.Sensor_2	1

D:\CIROS\RFAS\Library_Bosch_CTS40\Stopper\ST_Test-SimControl.mod
Modell-Explorer
CIROS Studio (nur für Aus- und Weiterbildungszwecke)
E/A-Verbindungen

Fig. 6-24: V&V of components – Stopper unit

The measurement range of sensor 2 has been elongated from 7mm to 70mm (red box with sensor property window) so that the box is easier to move into the sensor beam. At the bottom, the activation of sensor 2 by the box is indicated with orange marker. The manual switching of the stopper block leads to a correct status change of Transport Simulation node (cyan marker).

Transport car

Due to its complex behaviour, the V&V of the transport car needs special attention. The logic from the conceptual model, which has been validated using the real transport car, has been translated manually to a program for the logic controller inside the component model, which tends to be an error-prone task.

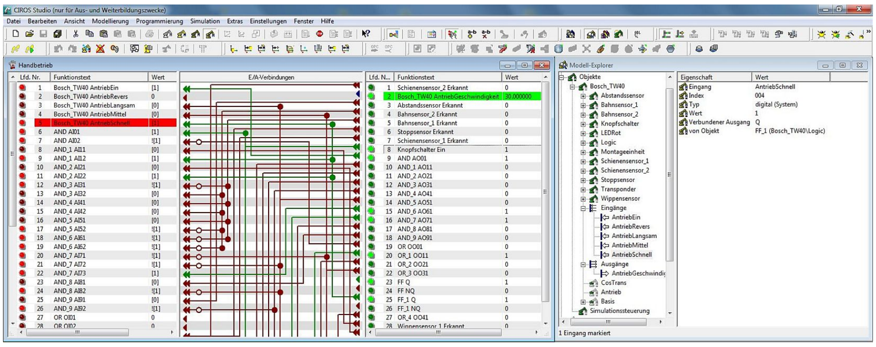


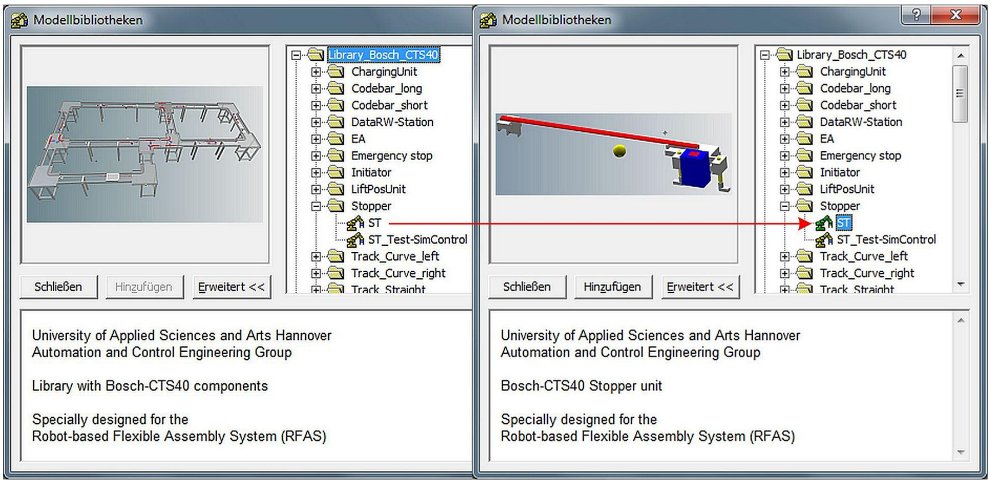
Fig. 6-25: V&V of transport car logics

After adding a SimController, the V&V of logics implemented in CIROS can be done by manual operation (Fig. 6-25). It is possible to activate e.g. sensor inputs and view the reaction of outputs such as resulting speed. The results have to be compared with those from conceptual model.

After completed V&V of component models, they can be stored in a library for reuse during High-level Plant Modelling.

6.1.2 High-level Plant Modelling of the transportation system

The High-level Plant Modelling comprises the selection of component models from the specifically designed Bosch CTS40 library (Fig. 6-26) and its composition to the entire transportation system.



University of Applied Sciences and Arts Hannover
Automation and Control Engineering Group

Library with Bosch-CTS40 components

Specially designed for the
Robot-based Flexible Assembly System (RFAS)

University of Applied Sciences and Arts Hannover
Automation and Control Engineering Group

Bosch-CTS40 Stopper unit

Specially designed for the
Robot-based Flexible Assembly System (RFAS)

Fig. 6-26: CIROS library for Bosch CTS 40 components

The right part of figure 6-26 shows the stopper unit which is used nine times in the entire system (once as part of the lifting-positioning unit). All necessary components for the entire transportation system are depicted in figure 6-27. The display of nodes and segments of the Transport Simulation has been switched off.

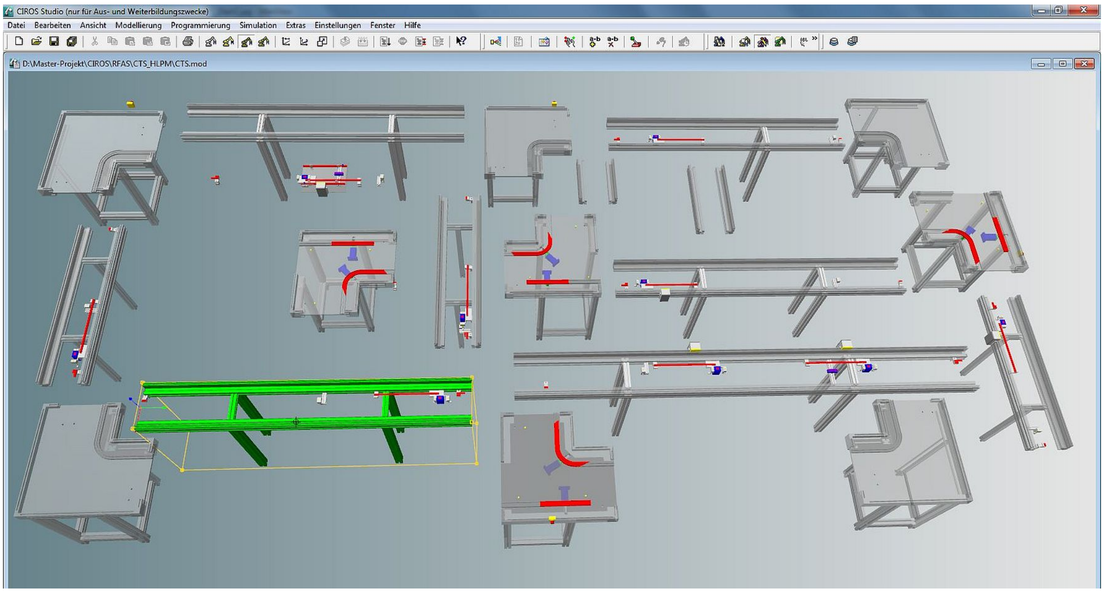

CIROS Studio (nur für Aus- und Weiterbildungszwecke)
Datei Bearbeiten Ansicht Modellierung Programmierung Simulation Extras Einstellungen Fenster Hilfe
D:\Master-Projekt\CIROS\RFAS\CTS_HLPM\CTS.mod

Fig. 6-27: All components of transportation system

This placing off all components at once has been done for clarification only. Generally, the High-level Plant Modelling is done sequentially by placing and connecting one component model after each other.

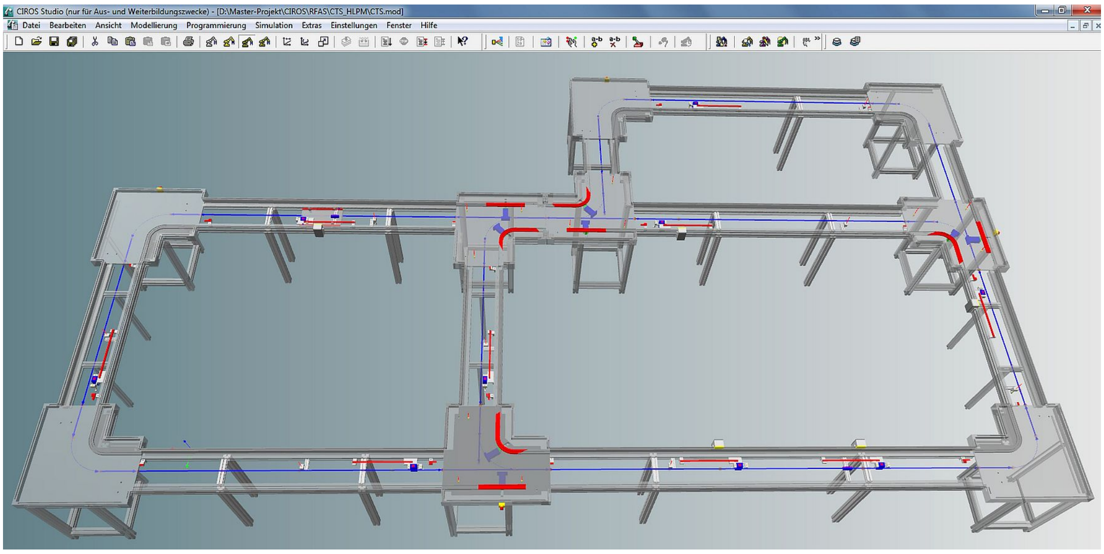

CIROS Studio (nur für Aus- und Weiterbildungszwecke) - [D:\Master-Projekt\CIROS\RFAS\CTS_HLPM\CTS.mod]
Datei Bearbeiten Ansicht Modellierung Programmierung Simulation Extras Einstellungen Fenster Hilfe

Fig. 6-28: Entire transportation system after composition from component models

Figure 6-28 shows the transportation system after correct placement of all component models. The visualisation of the blue segments and nodes of the Transport Simulation has been switched on again.

6.1.3 Realisation of simulation environment for VC

As first step of the projected environment (cf. Fig. 5-5), the coupling of the HMI (WinCC Flexible Runtime), the virtual PLC (S7-PLCSIM) and CIROS with the 3D simulation model of the transportation system has been realised running on one PC. The software requirements regarding e.g. the compatibility of software versions to cooperate has been checked. For example, the direct coupling of CIROS is only possible with S7-PLCSIM 5.4 SP3 or newer. In spite of having the correct versions, initially the coupling does not work until a dll file inside CIROS, provided by the RIF, has been exchanged by a newer one. First the coupling between S7-PLCSIM and the runtime version of WinCC Flexible does not work. It turned out, that an additional server licence for WinCC Flexible is necessary for this purpose. After purchasing this licence from Siemens and installation on the PC, the coupling between S7-PLCSIM and WinCC Flexible is achieved.

7 Validation

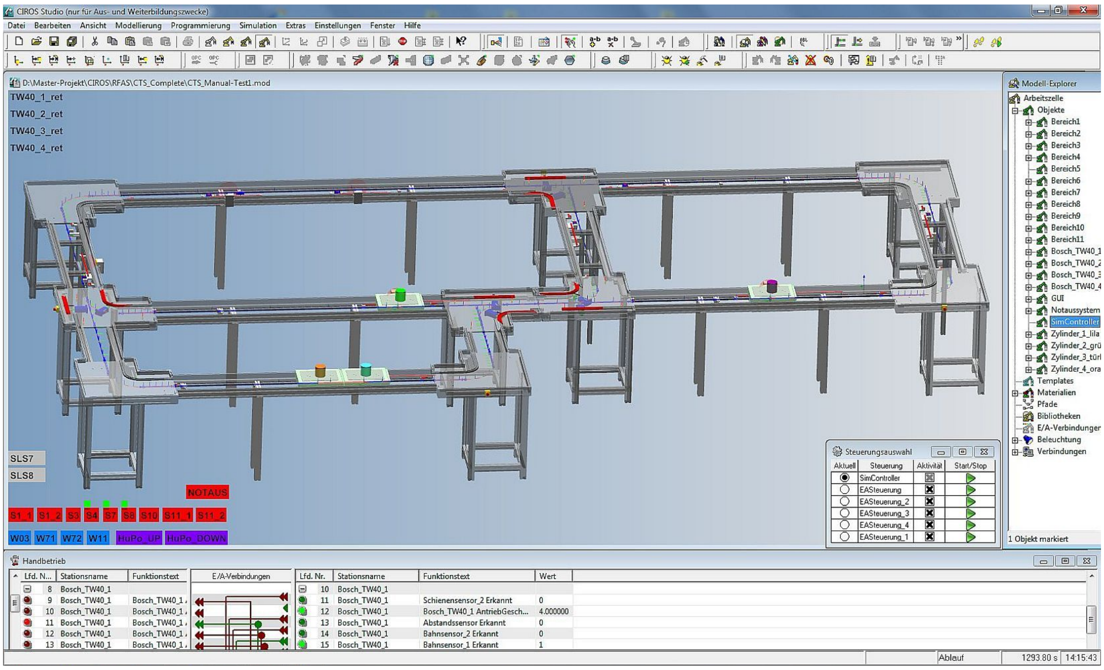
The V&V of the plant model should not be conducted before the V&V of the component models has been successfully completed. Generally, only intensely tested models should be included into a simulation model library. Even after an intensive V&V procedure of component models, it is possible that these models still show errors or irregular behaviour when different component models are interacting because not all situations have been considered before. Nevertheless, well-tested component models alleviate the error localisation during V&V of the plant model, because then occurring errors can only originate from wrongly built system during High Level Plant modelling (e.g. wiring error) or from faulty component model / sub-system interaction.

7.1 Verification and Validation of plant model

The proposed procedure for High-level Plant Modelling and VC (Subsection 5.3.5, Fig. 5-18) recommends a verification and validation after completed modelling. The simulation model has necessarily to pass through a V&V procedure to avoid errors as mentioned above. Otherwise, the VC analyst would have problems to distinguish modelling errors from errors of the PLC program or mechanical design during VC. Furthermore, it is necessary to find out, as much as possible, if all potentially erroneous situations of the system as a whole have been considered during modelling and consequently whether the simulation model is able to react to errors of controller programs in an adequate manner. Thus, the question is if the VC analyst will be able to recognize program errors by reactions of the simulation model. A typical example is the indicating of mechanical collisions of selected components by e.g. colour change or simulation stop. Another example is the checking of sure sensor activation by review of mechanical fit and correct sensor measuring distances.

In the end, the question arises if the simulation model behaves like the real system. In this thesis, it has been possible to compare model and hardware but generally, the modelling for VC has to be done without having the hardware available, which is a problem for a trustworthy validation.

For a convenient handling of the simulation model during V&V procedure it is helpful to use the internal "SimController" of CIROS supported by a small GUI, specifically designed for the RFAS, to allow manual operation without using or having a PLC program (Fig. 7-1).



Steuerungsauswahl

Aktuell	Steuerung	Aktivität	Start/Stop
●	SimController	☒	▶
○	EASteuerung	☒	▶
○	EASteuerung_2	☒	▶
○	EASteuerung_3	☒	▶
○	EASteuerung_4	☒	▶
○	EASteuerung_1	☒	▶

Handbetrieb

Lfd. N...	Stationsname	Funktionstext
8	Bosch_TW40_1	
9	Bosch_TW40_1	Bosch_TW40_1
10	Bosch_TW40_1	Bosch_TW40_1
11	Bosch_TW40_1	Bosch_TW40_1
12	Bosch_TW40_1	Bosch_TW40_1
13	Bosch_TW40_1	Bosch_TW40_1

E/A-Verbindungen

Lfd. Nr.	Stationsname	Funktionstext	Wert
10	Bosch_TW40_1		
11	Bosch_TW40_1	Schienensensor_2 Erkannt	0
12	Bosch_TW40_1	Bosch_TW40_1 AntriebGesch...	4.000000
13	Bosch_TW40_1	Abstandssensor Erkannt	0
14	Bosch_TW40_1	Bahnsensor_2 Erkannt	0
15	Bosch_TW40_1	Bahnsensor_1 Erkannt	1

Fig. 7-1: Manual operation of simulation model for V&V using CIROS SimController

In the case of the used transportation system, the interacting of track components and transport cars has to be checked carefully regarding geometrical fitting, especially related to sensor interaction. On the one hand, virtual sensors inside track models are activated by the transport car models and on the other hand, the virtual sensors of the transport car have to be activated by track models (code bars, signal bar) and by its environment (other transport cars, tracks). Some examples are following below.

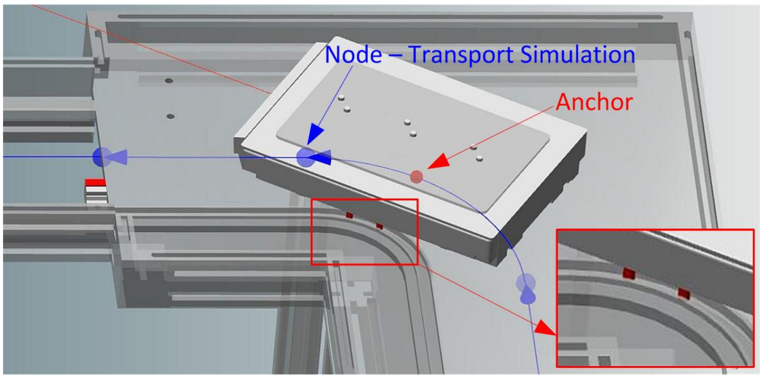


Fig. 7-2: V&V of plant model – Transport car in curve

The transport car moves with its anchor (red bullet) along the track segments of Transport Simulation (cf. subsection 5.2.1) from node to node (blue bullets) following the guiding track of the curve. The track idlers of the transport car are exactly inside the guiding profile (Fig. 7-2).

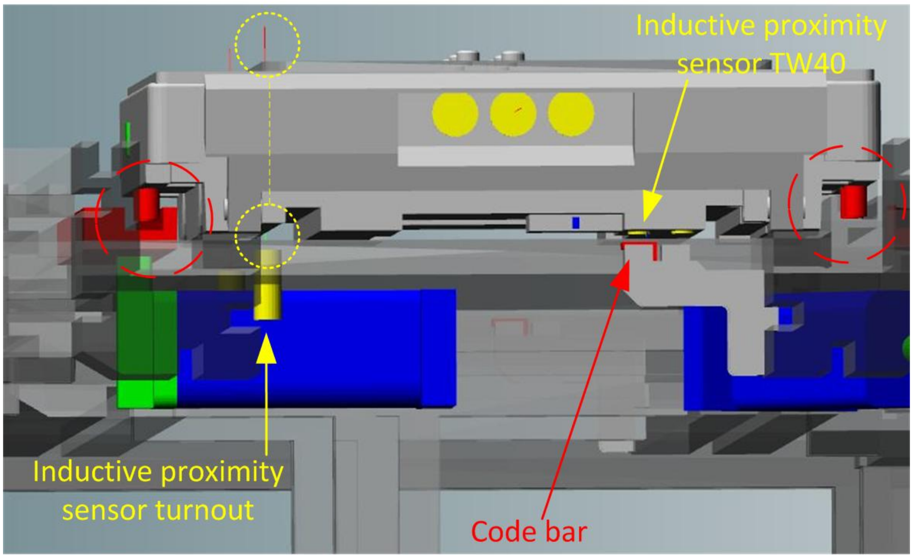

Inductive proximity sensor TW40
Inductive proximity sensor turnout
Code bar

Fig. 7-3: V&V of plant model – Transport car in turnout

In addition, when driving through a turnout, the track idlers of the transport car move exactly inside the guiding profile (Fig. 7-3, red circles), both inside the moveable profile (red) and in the static part. The inductive proximity sensor of the transport car is well positioned above the code bar. The measurement range of the inductive proximity sensors of the turnout has been adjusted too long (yellow circles) which needed to be corrected, because only the relevant part of the underbody of transport car has to be detected.

The following examples show some different erroneous states that should not occur when the system is controlled by a correct PLC program. During V&V with manual operation, this has been induced intentionally to see the simulation model reaction.

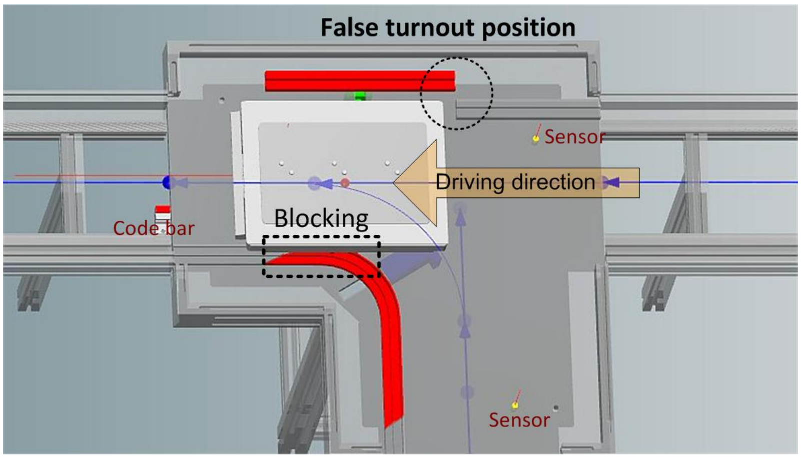


Fig. 7-4: V&V of plant model – Transport car blocked in turnout

The transport car drives into a turnout in false position, not in straightforward position, as it would be actually required. The transport car stops correctly in the simulation, blocked by the curved guidance profile (Fig. 7-4) which has been made possible by an additional sensor in the transport car. Otherwise, the transport car would simply pierce the geometry of the curved profile and move on, unless this would be prevented by a collision calculation and detection with its computational load. The Transport Simulation detects only contacts or collisions of carriers among each other.

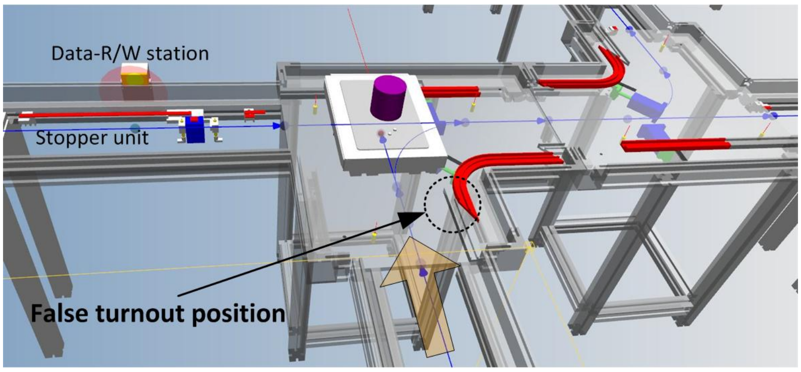


Fig. 7-5: V&V of plant model – Transport car stops in false turnout position

The false turnout position (correct would be right turn) leads to a correct stop of the transport car on opposite side, realised by a dead-end modelled in Transport Simulation (Fig. 7-5). This event does not affect the other transport cars, the simulation continues running.

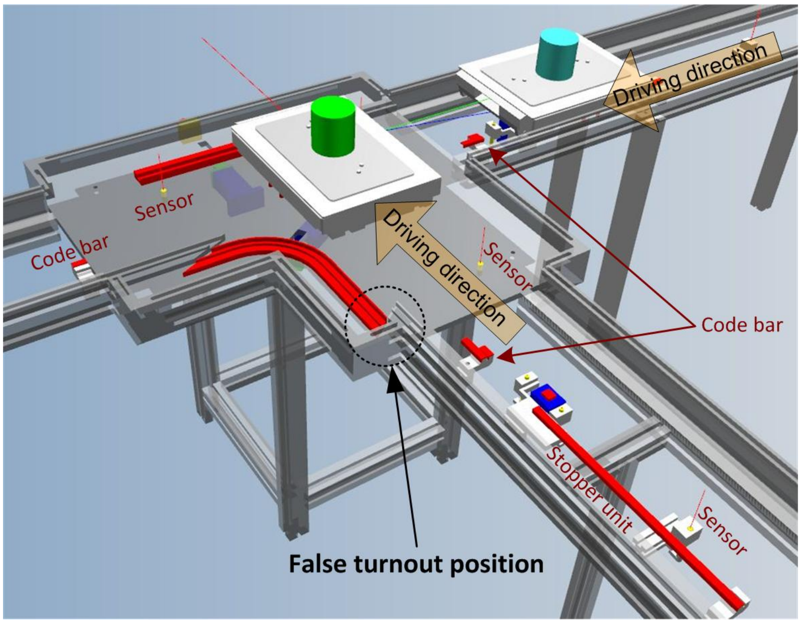


Fig. 7-6: V&V of plant model – False turnout position stops transport car

A false turnout position (correct would be left turn) leads again to a correct stop of transport car (green) on opposite side. Due to the still running simulation, it would be possible to release the stopper and the other transport car (cyan) would crash into the first one (Fig. 7-6).

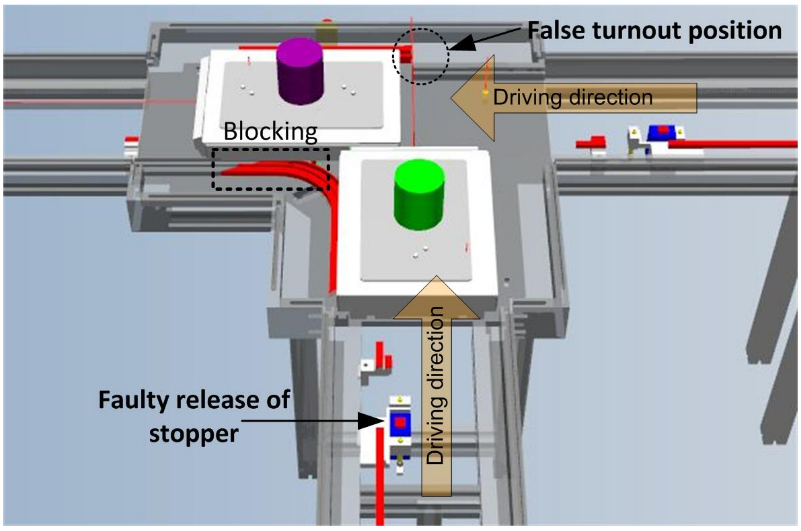


Fig. 7-7: V&V of plant model – Two transport cars crash in turnout

The first transport car (violet) has been blocked to stop because of the false turnout position. A faulty release of the stopper let the second transport car (green) crash into the first one (Fig. 7-7).

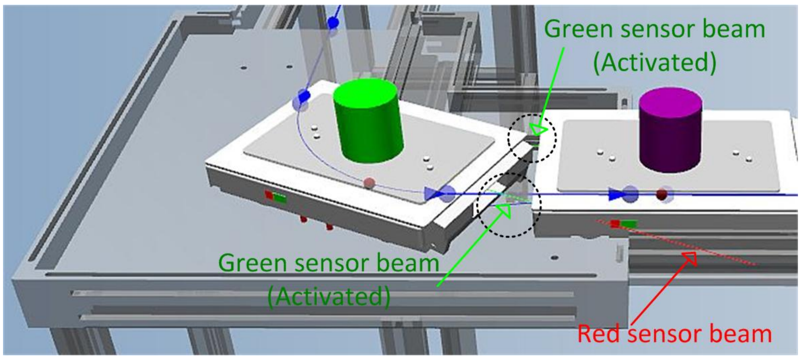


Fig. 7-8: V&V of plant model – Crash of two transport cars in curve

In this case, the rear transport car (green) has been stopped correctly by activation of its front rocker sensor when driving against a standing transport car (violet). The green sensor beam of the optical sensor shows that the obstacle has been detected, but the logic of the transport car ignores this sensor inside curves or turnouts (Fig. 7-8). Thus, the transport car drives against the other transport car with its curve speed and not with minimum speed. Such a crash (contact with speed faster than minimum) has to be prevented necessarily in reality.

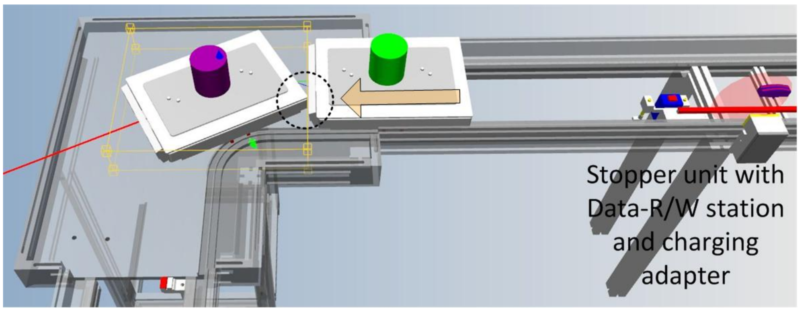


Fig. 7-9: V&V of plant model – Crash of two transport cars in curve

The first transport car (violet) has been switched off inside curve (sensor beams are visible anyway). After releasing the stopper, the second transport car (green) crashes into the first one and stops simulated correctly (in this case by Transport Simulation). The second transport car is already inside the curve and driving with its curve speed and not with minimum speed (Fig. 7-9). This has also to be prevented necessarily in reality like the case above.

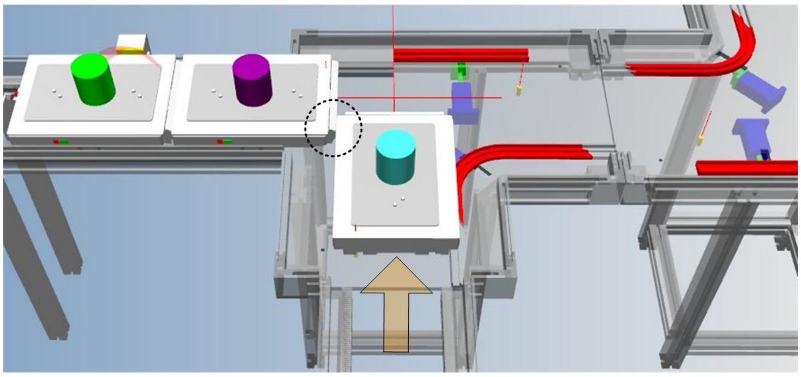


Fig. 7-10: V&V of plant model – Crash of two transport cars moving into turnout at once

Two transport cars (cyan and violet) move into the turnout at the same time (Fig. 7-10). The simulation stops both transport cars when collided. This is also a critical situation to be prevented in reality, because both transport cars (cyan and violet) are moving with curve speed. This could result in hardware damage of the front rockers. The third transport car (green) contacts and stops at regular minimum speed.

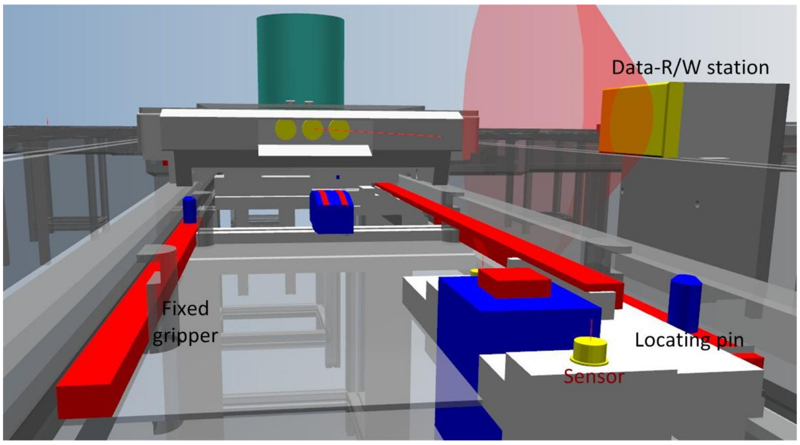


Fig. 7-11: V&V of plant model – Transport car moving into lowered lift-positioning unit

The transport car (Fig. 7-11) is moving correctly simulated into the lowered lift-positioning unit.

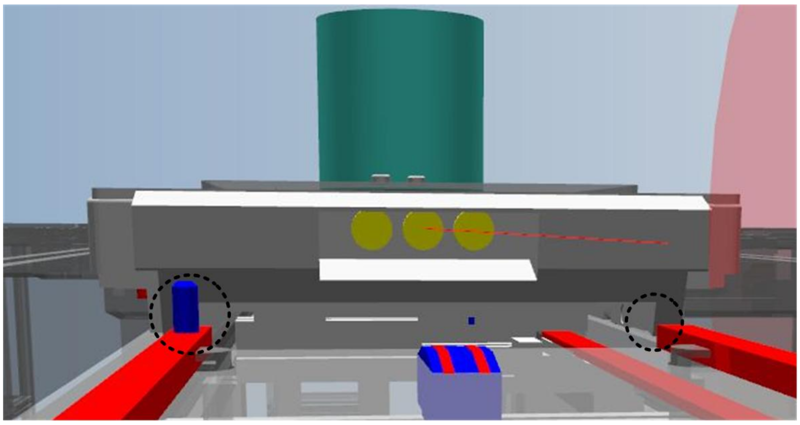


Fig. 7-12: V&V of plant model – Crash with raised lift-positioning unit not detected by simulation

This is an example for an erroneous situation that has not been considered during modelling. The lift-positioning unit inside the SCARA robot cell is in the raised position and the transport car is allowed to drive into the cell (Fig. 7-12). In reality, this would result in a crash, in the simulation the transport car simply pierce the geometry of the lift-positioning unit and move on. The Transport Simulation only detects contacts or collisions of carriers and the model of the transport car has now additional sensors to detect this crash. Here a remodelling or the activation and configuration of collision calculation and detection become necessary.

Additionally a tuning of model parameters such as speeds of transport cars is possible and recommended to approximate the behaviour of virtual system to that of the real system.

After having completed the V&V of the simulation model of the entire transportation system including transport cars using CIROS internal simulation controller for manual operation, the next step has been the VC.

7.2 Virtual Commissioning of RFAS transportation system

Based on the configuration projected in subsection 5.2.3 (Fig. 5-5) a VC has been conducted. This VC, carried out in a SIL simulation on one PC (Fig. 7-13), incorporates:

- Simulation model of CTS40 transportation system, with same layout as hardware system as part of the RFAS, running inside CIROS including:
    - Transport car models with internal logics controller interacting with the CIROS Transport Simulation
    - Data-R/W station models with CIROS I/O controller executing IRL programs to simulate the data exchange between mobile data carriers of transport cars and PLC
- S7-PLCSIM, the emulated PLC is executing the original STEP7 program for the control of the real transportation system including communication with Data-R/W stations.
- WinCC Flexible 2008 Runtime, the HMI, running as runtime version on the PC, allows the operation of the simulated transportation system by using the original control displays intended for the HW operator panel.

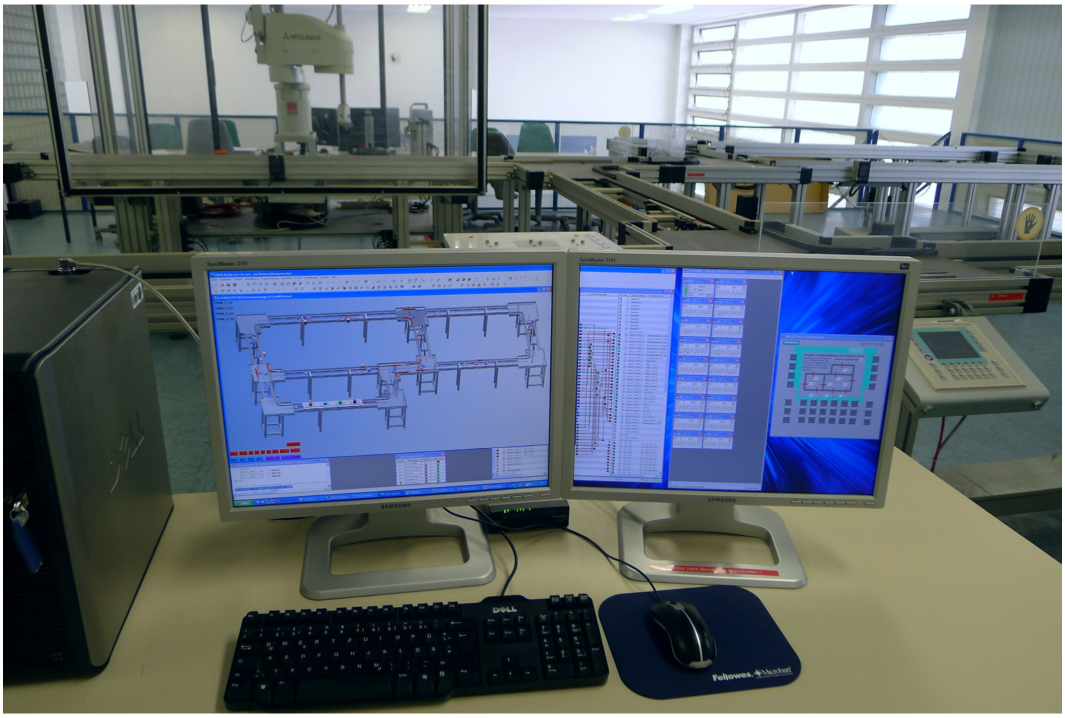


Fig. 7-13: VC of CTS transportation system in SIL configuration

Figure 7-14 shows a screenshot from above VC setup. The operation of the simulated system is possible by pressing the simulated membrane keys or buttons in the touch area of the simulated panel or if applicable in the STEP7 program by selecting input bits in S7-PLCSIM.

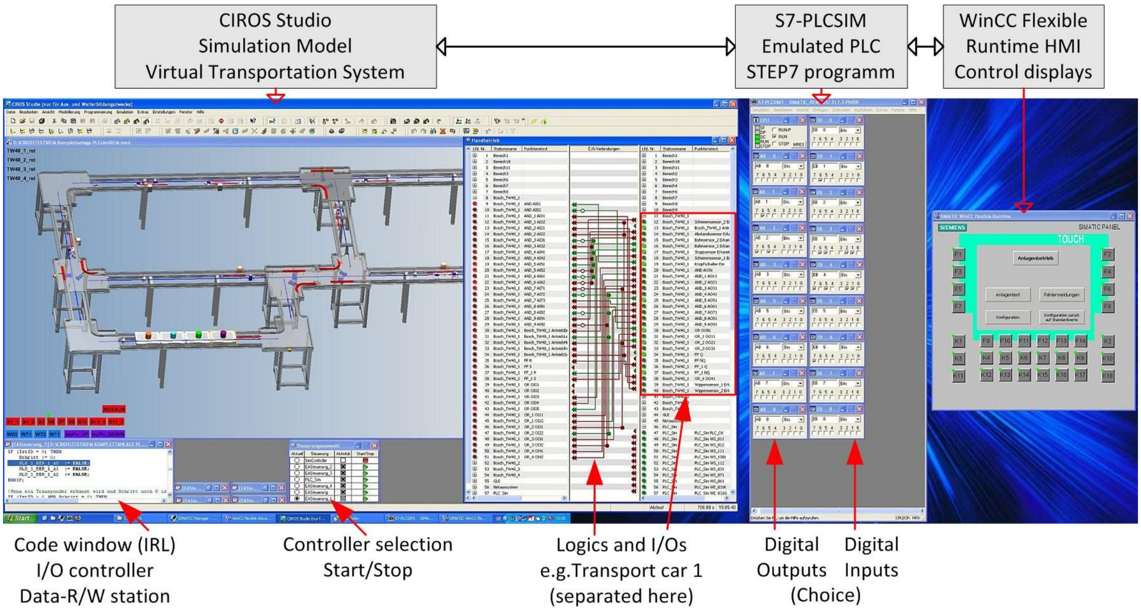


Fig. 7-14: Screenshot (both screens) – VC with CIROS, PLCSIM and WinCC Flexible Runtime

It is also possible to watch e.g. the I//Os of the transport cars during simulation, as indicated in Fig 7-14 with the separated area for transport car 1 (red rectangle). The VC analyst can for example observe which sensors are activated ("LEDs" lime-green) and which resulting speed the transport car just has. The controller selection window allows include/exclude and start/stop controllers. The code window shows the cyclic running IRL code from the I/O controllers of Data-R/W stations. The VC begins with test of manual mode (selected via HMI), with activation of e.g. stoppers and turnout in the control displays by pressing the simulated membrane keys and observing the correct actions of the correspondent components in simulation model. The implemented interlocking functions of the PLC program are also tested. They have to prevent such situations as e.g. depicted in the validation of the simulation model (see above). Afterwards, the automatic modes are selected via HMI and the correct travelling of transport cars can be observed.

This successful implementation shows that it is possible to conduct a VC in a SIL configuration with CIROS Studio coupled to PLCSIM (executing an original STEP7 program) and original control displays with the runtime version of WinCC Flexible as hypothesised in subsection 5.2.2.

8 Discussion of results

In this chapter, the results gained from addressing the research objectives are summarized considering the conducted research based on the test-bed and possible drawbacks are discussed.

A systematic approach for SMEs to facilitate the substantially high modelling effort required for VC of manufacturing systems has been defined as main aim of this research. For this purpose, the entire simulation model building process has been investigated and systematic methodologies have been developed.

The research objectives, defined in subsection 1.2.2, and how they have been addressed in this thesis is described below:

1. To understand the state of the art in VC of mechatronic manufacturing systems
2. To determine the requirements for Virtual Commissioning
3. To identify the existing approaches to set up an environment for VC
4. To understand the simulation model building and its challenges and how to address them
5. To develop viable procedural methodologies that can facilitate the aims of the study
6. To validate the procedures developed in such a way as to indicate the feasibility and desirability of Virtual Commissioning in the engineering environment described

Objective 1: The investigation to understand the current state of VC started with a literature review. This review investigates the underlying technologies such as mechatronics, simulation, VR etc. and reveals principal advantages and the potential of VC, documented by several examples stating e.g. reduced real commissioning time, higher quality planning and better control software quality. In spite of the undisputed advantages, the utilisation of VC, especially in SMEs, is not as common as it is to be expected. One main reason for the rare use of VC, besides its general complexity, is a high modelling effort (related to objective 4) to build the necessary virtual plant models. Modularisation and the use of simulation model libraries with mechatronic component models have been identified as promising general approaches to address these difficulties.

However, from existing literature no evidence of methodologies for systematic implementation and reasonable execution of VC have been found. This current lack of availability of such methodologies will also be addressed by the VDI/VDE-GMA technical committee FA 6.11 for different use-cases (process industry, discrete manufacturing industry, diverse sizes of enterprises etc.) still to be determined during future work. Hence, one goal of this thesis has been the development and provision of procedural methods applicable for SMEs (related to objective 5).

Objective 2: The general requirements for VC have been analysed, and besides some principal and organisational issues, the data requirements as well as hardware and software tool requirements have been identified. Besides their utilisation for the VC in this thesis, the results of this investigation have been introduced into the developed procedural methods.

Objective 3: Based on the general requirements for VC (objective 2) and the particular requirements derived from the hardware system used as test-bed in this thesis, the requirements for the VC environment in this thesis have been specified. From the variety of existing approaches to set up a configuration for VC (simulation tools, coupling to real/virtual controllers etc.), an appropriate 3D plant simulation system (CIROS Studio) has been chosen and an environment appropriate for the VC has been designed and configured, according to this specification. CIROS Studio has been integrated in this environment together with Siemens S7-PLCSIM, which has been used as virtual controller for the VC in a SIL configuration, but the environment has been designed to be suitable for future work and provides the ability to integrate other controllers.

Objective 4: The simulation model building is the crucial task in VC and has been investigated very intensively. The development of mechatronic component models, including the preceding CAD preparation, has been identified as major challenge in the entire simulation model building procedure and as the underlying reason for the complaints of high modelling effort.

The use of 3D CAD models for simulation model building typically shows three drawbacks: the CAD models are too complex, unsuitably structured and the kinematics almost always represent a problem. This thesis provides solutions how to handle the complexity and the unsuitable structure of CAD models for VC in a CAD preparation procedure. The transfer of 3D CAD models with kinematics from CAD to simulation tools is not possible in the vast majority of cases. Neither the import of native CAD formats supporting kinematics in simulation tools, nor the principal kinematics transfer possibility of formats such as STEP or COLLADA has been implemented in tools, apart from very few exceptions. It is proposed in this thesis, that this has to be solved by further developing CAD exchange formats and more advanced CAD tool export and simulation tool import possibilities at long sight.

The modelling has been simplified by the use of the Transport Simulation, which provides a guidance of the transport cars by defining segments and nodes in the tracks. A kinematical modelling of the transport cars routed by the guidance profiles would have been very difficult. Thus, such a modelling feature is advantageous for carrier-based transportation systems. In the case of packaged goods or e.g. bottles on conveyors a simulation tool providing physical modelling would be recommended, because the use of the Transport Simulation as well as kinematical modelling of each package would be too laborious or in the case of e.g. hundreds or thousands of bottles impossible. Besides, accumulating effects etc. can only be modelled in a reasonable manner with physical simulation.

Objective 5: The first developed methodology comprises the implementation and execution of VC in its entirety (cf. Fig. 3-1), but is reduced to the essentials to give an overview of the whole process. This methodology can be divided into the three main phases: conception, implementation and analysis. The conception phase includes the steps Problem Definition/Cost Analysis, Project Planning, System Definition and Model Conceptualisation and finally Data Collection and Preparation. With the exception of the first step (Problem Definition/Cost Analysis), this thesis addresses all steps in detail. In Project Planning, the focus is on the tasks: requirements for VC, specification of VC environment as well as on the selection of a suitable 3D plant simulation tool including the selection of simulation method (SIL/HIL) and communication (these tasks are related to objectives 2 and 3). The System Definition and Model Conceptualisation in this thesis are based on a decomposition of the real manufacturing system to allow a modular, component-based modelling approach.

The implementation phase is formed by the simulation model building including model verification and model validation (related to objective 4). Due to the advantages of the modular, component-based approach (possible drawbacks see below), the simulation model building procedure has been divided into two methodologies for the separate modelling tasks “Low-level Component Modelling” (Fig. 5-9), conducted after decomposition of the real manufacturing system, and “High-level Plant Modelling” (Fig. 5-18).

Due to a thoroughly conducted CAD preparation, an unimpeded Low-level Component Modelling was made possible. The component model of the transport car has been simplified, the battery charging and sleeping modes of the car have not yet been implemented. Because of the comparatively small track layout and two charging units of the real system, this is dispensable for the VC of the RFAS, but it is a challenging modelling task for the future work.

The analysis phase mainly comprises the specification of test cases/scenarios for VC, the execution of VC and the documentation of simulation results. In industrial applications of VC, it is standard to define the test cases manually, as in this thesis. If there are special needs to test the reaction of control programs to all possible combinations of input signals this could result in thousands of cases, which have to be tested (Naake et al., 2012). The authors propose to implement test automation and applied a test tool from automotive ECU testing to VC. This would also head for Functional Safety (subject of several regulations such as IEC 61508, IEC 61511, IEC 62061), but a possible relation of VC and the securing of Functional Safety, especially for safety-critical systems, according to regulations has not been considered in this thesis. Thus, in this thesis the VC of the considered transportation system has been conducted based on a manual test design, mainly by running manual mode and automatic programs and observing the virtual system if the process runs correctly.

Objective 6: The validation of developed methodologies (Figures 3-1, 5-9, 5-18) and procedures (CAD preparation, Low-level Component Modelling and High-level Plant Modelling) has been done using the transportation system, the major subsystem of the real manufacturing system at the UASA Hannover, as test-bed.

8.1 Discussion of CAD data preparation as basis for Low-level Component Modelling

The CAD data preparation has been investigated in detail and simplification and component structuring have been identified as decisive tasks. The simplification addresses the complexity of CAD models by object filtering (removing or exchanging parts), feature suppression (remove features or reduce the detailing of features such as holes) and tessellation/mesh simplification. The crucial component structuring, adjusts the inner structure (parts, assemblies) of CAD models and separates static units, moving units and sensors/communication units. A thoroughly conducted CAD preparation eases the following Low-level Component Modelling and thus reduces the effort for simulation model building significantly, but is itself a laborious task.

This laborious CAD data preparation, following the workflow outlined in figure 5-19, would always necessarily be repeated for different simulation tools and/or different users (even if using the same tool) to build CAD components suitable for later Low-level Component Modelling for VC. Consequently, this CAD preparation should better be done once by CAD specialists of the component vendors as outlined in figure 5-20. If the manufacturers of components and subsystems would generally provide in the future 3D CAD models suitable for dynamic simulation within VC, which is already possible today in principle, this would be a good basis for the proposed future provision of mechatronic models (cf. Fig. 5-21).

The CAD data preparation has been applied to the component models in this thesis, and the expected positive effects could be validated by modelling in CIROS. The experiences with CAD data transfer between CAD tools (Solid Edge, Creo Parametric) and simulation tool (CIROS) in this thesis show that not only kinematics transfer is problematic, but even the transfer of material properties or correct colours is not assured. Such problems should be addressed by tool suppliers, because data such as the material properties can be important for VC (e.g. for detection by simulated inductive sensors).

8.2 Discussion of the modular component-based approach for simulation model building compared to a non-modular approach

The disadvantages of a non-modular approach such as ineffective and exceedingly complex modelling as well as the advantages of the proposed and selected modular approach such as reusability, flexibility and testability have been presented in subsection 5.3. A reuse of designed component models for a new installation of a CTS transportation system was not realized in this thesis, but opens the path to the flexibility. If it becomes necessary for a company to redesign e.g. the layout of an existing transportation system due to changed or additional subsystems such as robots or machining tools, this can be done flexibly and easily based on an existing component model library. A simulation model design without decomposition of the entire system would not allow this respectively would raise so much effort that maybe a new simulation model building would become necessary. A small example for a new layout, composed of component models from designed CTS library, is given in figure 8.1.

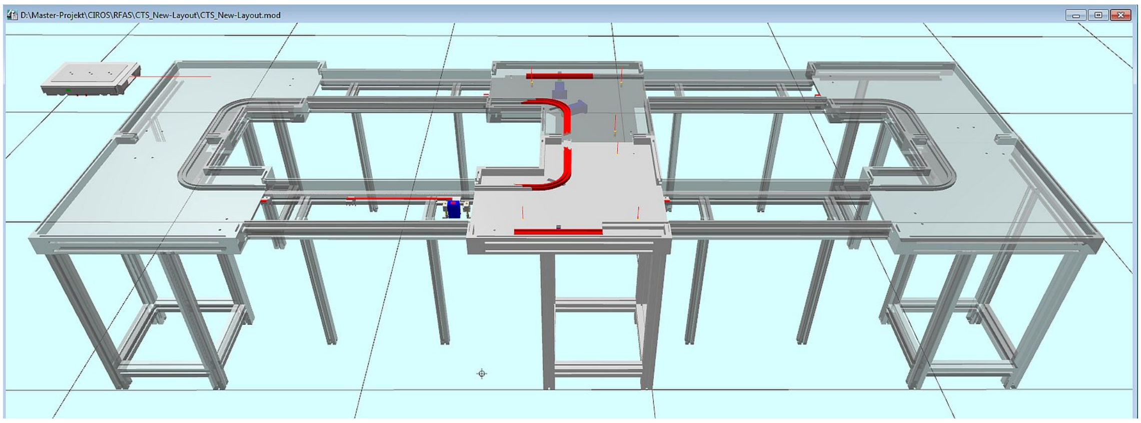

D:\Master-Projekt\CIROS\RFAS\CTS_New-Layout\CTS_New-Layout.mod

Fig. 8-1: Example for alternative layout based on CTS component library

The usability of this flexible modelling can be enhanced by providing a snapping functionality to the track components as possible in simulation tools such as CIROS or Experior. The modelling of other subsystems such as robot cells can also benefit, thus it is recommended to use a component-based approach also for e.g. the periphery in robot-cells such as gripper stations. Hence, it will be used for the modelling of the robot cells inside the RFAS during future work.

How one can take advantage of a good testing methodology has been demonstrated for track components and the transport car model. All track components as well as the model of the transport car have gone through a V&V procedure, as exemplarily presented in subsection 6.1.1.5, before they became part of the CTS library.

An additional advantage of the proposed modular component-based approach and the consequent implementation of splitting the modelling task into Low-level Component Modelling and High-level Plant modelling is the thereby possible wider usage of VC in a company. The modelling and simulation expert is still necessary to build the component models (Low-level Component Modelling) as long as the novel workflow with tool-independent provision of mechatronic models for standard component by suppliers (cf. Fig. 5-21) has not been realised. For purpose-built components, a company simulation expert will potentially be still necessary in the future, unless also supplied purpose-built components come with component models. However, following the proposed approach, this expert will not any longer do the complete development of the virtual manufacturing system and implementation of VC, instead he will do only the Low-level Component Modelling to complement the simulation model library with new component models. The comparatively easy High-level Plant Modelling and the execution of VC can be then done by several trained simulation users.

Besides undeniable advantages, the modular component-based approach has some possible drawbacks. These are partly the same as in mechatronical design of reusable real components. Thramboulidis (2008) states that available components might be difficult to integrate into a new system. This is a typical problem in the design of reusable components, not only for real or virtual mechatronic components but also e.g. in software development. The smart definition of boundaries and accordingly the level of granularity for mechatronic components is a challenging task, subject of ongoing research in the area of mechatronics (Weyrich et al., 2014), but still also depending on the experience of the developer. An important factor for the successful reuse is the level of granularity, but it is difficult to decide at which level of granularity the models will be best designed to be reusable (Maga et al., 2011). From different definitions of granularity in literature, Maga and colleagues propose to define it in the context of modelling by the proportion compared to the entire system (a work cell is on a lower level of granularity than a plant section) and by the level of detail, more details result in finer granularity. Furthermore, Maga and co-workers refer to a conflict regarding granularity. The best flexibility, maintainability and reusability is provided by fine-grained models (being least efficient), whereas coarse-grained models provide the best efficiency, being less flexible and reusable. This is also illustrated by the pyramid with hierarchical levels for simulation model building of the RFAS (see Fig. 5-6). The basic, fine-grained component models at the bottom are highly reusable, flexible, maintainable and testable, but the simulation model building is less efficient than with more coarse-grained models at higher levels. Contrariwise, models of e.g. track sections designed for the RFAS are efficient to use for this system, but might be not appropriate for another manufacturing system. Weyrich et al. (2014) state that *“modules should be large and functionally comprehensive”* because then they *"can be quickly and easily used for the creation of new simulation models"* but they put into perspective that *"too large modules are complicated to scale and modify"*. Maga et al. suggest, *"to use well-documented, hierarchical, nested models, which provide different levels of granularity depending on the required functionality"*. Such a hierarchical, nested structure cannot be represented with the simulation model library in CIROS that provides only a flat structure, thus it would only be possible to store all basic components and additionally functional groups etc. (e.g. single stopper, sensors and signal bar and altogether as entire stopper unit etc.). However, doing so would possibly result in another drawback identified by Thramboulidis, a maybe exponentially rising number of components in real companies and difficulties for the modeller to find the proper ones. The number of components being necessary for the considered transportations system and even the entire RFAS is manageable and the granularity is partly predefined by the hardware catalogue, but for other manufacturing systems, further research on definition of granularity and detailing of component models for VC as well as their handling in libraries seems to be reasonable.

Additionally, Thramboulidis refers to another possible drawback, if component models need to be modified, e.g. due to changes of the real component, the benefit of having components already verified and validated will be reduced. As a result, effort will arise through a necessary library management (versioning), which is currently not provided by CIROS and comparable tools, to keep the models up-to-date (with preserving the old models), at least before starting a new modelling project. For purpose-built models, this would be a task for the simulation expert, for standard components or other supplied components; this could also be addressed by the proposed future provision of mechatronic component models (cf. Fig. 5-21). Every time the supplier changes a hardware component, he would provide a modified component model.

Besides, in the case of one-time projects (no re-use by application to new systems) the modular component-based approach will imply slightly more effort compared to the approach without decomposition because the modularisation requires additional work. In this thesis, the advantages of decomposition and a modular approach, such as less complex modelling of the entire system, flexibility and testability (cf. subsection 5.3), are uprated beyond the possible drawbacks, even without future reuse for new systems.

8.3 Discussion of novel workflow for VC with provision of mechatronic component models

Based on the proposed modular, component-based approach for simulation model building, it has been shown that a new strategy for the provision of mechatronic component models (cf. Fig. 5-21) is principally possible and would be an advantageous step towards further eased development of virtual manufacturing systems for VC. The proposal to relocate Low-level Component Modelling to the component manufacturers could reduce the modelling effort for the VC simulation model building, thus the manufacturers have to be encouraged by customer requests to provide such models together with the hardware components, but indeed this is a chicken-and-egg question. A motivation might be the competitive advantage compared to suppliers not providing such models. The provision of appropriate component models would greatly reduce the costly Low-level Component Modelling always repeated by simulation experts in different companies using the same off-the-shelf components to a one-time activity for the component manufacturer.

For this purpose, a conceptual design for a mechatronic component model has been proposed (Fig. 5-22) that incorporates at least 3D geometry, kinematics, sensors, electrical I/Os, behaviour/logics and technical data in which 3D model and behaviour model are linked. The transport car, used in this thesis, is an appropriate and typical example of a complex off-the-shelf component or sub-system, for which ideally the manufacturer should provide a mechatronic component model. Presumably, the manufacturer would be best placed to know the complete behaviour, or at least has a better understanding than simulation users who would otherwise be forced to create models for themselves.

The conceptual model of this transport car (cf. Fig. 6-15), which has been taken as the basis for the programming of logics/behaviour inside CIROS, has been developed according to interpretations and assumptions from the vendor manual, a practise that tends to be error-prone. Several researchers (cf. chapter 2) strongly suggest the validation of simulation models, preferably by comparison with the real hardware. In this thesis, it was possible to match the conceptual model, the logics model in CIROS and the behaviour of the real transport car against each other, but a VC has to be possible without having all hardware components available for modelling. If a reliable validation of a simulation model is not possible, errors appearing during VC are not explicitly referable to the control program or the real components, but might be errors of the simulation model.

The proposed mechatronic component model would provide a behaviour/logics model, which could be implemented e.g. in form of a PLCopen XML file (Fig. 5-23). Such provided behaviour model could be used as conceptual model in an IDE supporting PLCopen XML, such as the typical PLC IDEs MultiProg (KW-Software), logi.CAD (logi.cals) or 3S-Codesys V3 (PLCopen, 2011). The direct import of PLCopen XML into the logics controller of CIROS has not yet been implemented by the tool developers, thus a manual programming would be still necessary at this stage, but based on a provided model and not on interpretations and assumptions from the vendor manual. Another drawback of CIROS regarding this new approach is the unchangeable behaviour of kinematic mechanisms (cf. subsection 5.2.1), which can only be parameterised. For example, it is possible to define a speed for a push cylinder or speed and acceleration for a linear translation axis but not e.g. different ramps or a characteristic curve.

This mechatronic component model is also intended to provide the technical data to allow for automated parameterisation of sensors (measurement range etc.) and actuators (speed, stroke etc.). To date, this has to be done manually in simulation tools such as CIROS. Besides technical considerations, a drawback of this approach could be the reservation manufacturers might raise because they would be making public a large proportion of their commercially sensitive data of their products. This could be e.g. solved by providing only a simplified behaviour model. Such topics should be part of future research and joint efforts for standardisation of e.g. model content, levels of detail etc. A solution with implementation of "black-box" models could also be appropriate.

Furthermore, the manufacturers have the necessary product knowledge but probably often lack the modelling expertise to design mechatronic component models. Additionally, to implement such approach an easy-to-use editor to integrate all partial models is missing, in the case of AutomationML the editor should be able to integrate CAEX, PLCopen XML and COLLADA with 3D modelling, not only XML editing like the AutomationML editor.

The research conducted corroborates the supporting hypothesis of this thesis (cf. subsection 1.1). It is evident that it *is possible for SMEs to conduct Virtual Commissioning prior to full commissioning.* It is noteworthy that at the current state of art, an expert is still likely to be needed for Low-level Component Building. However, by following the recommended modular, component-based approach presented, the High-level Plant Modelling and VC can also be conducted by trained users in SMEs. The proposal for exchangeable mechatronic component models aims to supersede the need for expert model building locally in SMEs in the future, at least in most cases. The *debugging of control programs and control displays* is possible *and faults in the planned physical setup of the manufacturing system* can *also be detected and corrected.* An SME for validation of methodologies in an industrial environment has not been available, but based on experiences with planning, implementing and conducting a VC in an university environment with likewise limited capabilities as in SMEs, there is also evidence to suggest that the *VC approach would reduce fault detection and correction costs and efforts encountered during the commissioning phase of the real manufacturing system.*

9 Conclusions

This thesis presents a new proposal for a systematic simulation study methodology for Virtual Commissioning (VC), which is aimed at V&V of controller programs, HMI control displays and design of manufacturing systems by modelling and simulation. This methodology is intended to be notably beneficial for engineers from SMEs, being unversed in VC, as helpful guideline for planning, implementation and execution of VC.

Besides clarifying the requirements for VC, specifying an environment for VC and establishing the criteria to select an appropriate simulation tool, this thesis concentrates on the simulation model building as the crucial task for VC. Based on a proposal for modular, component-based simulation model building, this task has been split into Low-level Component Modelling, to be conducted for the components of the decomposed real manufacturing system, and subsequent High-level Plant Modelling of the virtual manufacturing system, necessary for VC.

The applicability of these new approaches has been successfully validated by planning, implementing and conducting a VC for a track-bound transportation system with self-driving transport cars, which is the major subsystem of a manufacturing system at the UASA Hannover, the so called "Robot based Flexible Assembly Systems" (RFAS).

The motivation for the research in this thesis came from the inadequate situation that existed regarding the V&V of collegiate programs for this RFAS. Before this research project, the V&V of the collegiate programs in this transportation system had not been possible without running the complete hardware system with its associated PLC and HMI systems, with all risks regarding damages being inherent in such a procedure.

The investigation of possible solutions has shown that the V&V of control programs for state-of-the-art industrial robots and machine tools is well supported by simulation, but ready to use solutions or methodologies for assembled manufacturing systems, e.g. composed of transportation systems and robot cells, are not publically available. This lack applies particularly to engineers in SMEs. They are confronted with considerable problems at commissioning when dealing with e.g. untested or partially tested control code for different interacting controllers, possibly inaccurately built systems and at worst, even faulty system design. In contrast to large enterprises from e.g. automotive industry, SMEs do not have the possibilities to address these problems by extensive implementation of the Digital Factory concept and its related application of complex suites of tools. Due to the still increasing complexity of industrial systems, the effort for commissioning tends to get progressively worse, and engineers in SMEs particularly, urgently need guidelines for planning, implementation and reasonable execution of VC to better the critical situation.

It turned out from literature as well as from discussions within the VDI/VDE-GMA technical committee FA 6.11, that such guidelines are currently missing. Thus, this thesis aimed at the development and provision of indicative procedural methods, to satisfy industrial needs, especially from SMEs.

The main novelties that resulted against this background from the research on VC so far are:

- A systematic simulation study methodology as general guideline for planning, implementation and execution of VC (Fig. 3-1). The basic procedural steps have been concluded from existing simulation study approaches, but specifically adapted to the needs of VC and successively filled with detailed content (specific tasks).
- The proposal for a systematic simulation model building procedure incorporating the systematic model building of mechatronic components (containing CAD preparation and Low-level Component Modelling) and High-level Plant Modelling (Subsection 5.3.3).
- Associated with these general methodologies, a procedure for systematic model building of mechatronic simulation components according to figure 5-9 and a procedure for High-level Plant Modelling and VC according to figure 5-18.

Based on these approaches, a novel workflow for Low-level Component Modelling has been proposed (Subsection 5.4). This proposal aims for the gradually relocation of this modelling task as far as possible to the origin of products, in the end the manufacturers should provide together with the deliverable components their mechatronic component models. For this purpose, a novel proposal for exchangeable mechatronic component models and a possible implementation with AutomationML (subsection 5.5) are outlined.

Besides, CIROS has been applied for the first time to a track-bound transportation system with self-driving carriers on passive tracks (Subsection 5.2.2).

10 Recommendation for future work

The simulation model of the transportation system should be tested with different collegiate PLC programs and HMI solutions in the future to gain more experience with VC and to find possible model errors or ‘nice to have’ enhancements. The modelling of the transport car TW40 has been conducted with several simplifications compared to the real transport car, thus an improvement of this model e.g. by simulation of battery lifetime and charging times or by modelling of kinematics for the front rocker would bring this model yet closer to reality.

The VC of the entire RFAS, incorporating transportation system, HMI and both robot cells, is one goal of the ongoing VC research at the UASA Hannover. For this purpose, it will be necessary to finally integrate the gantry robot cell and the SCARA robot cell into the CIROS simulation model. As justified in this thesis a gradual approach for the VC of the RFAS has been chosen. The suitability of the developed methodology has been successfully validated with simulation model building and VC of the transportation system. Now individual VC projects for each robot cell following this approved procedure are projected. After a successful VC of a sole robot cell, this robot cell can be integrated into the entire simulation model.

A HIL simulation is projected for the separate VC project of the gantry robot (cf. Fig. 5-5, indicated by orange boxes), but a possible SIL simulation using the CoDeSys Soft-PLC (PLCWinNT) instead of the motion controller TLM2 should be tried out too. The red boxes in figure 5-5 indicate the separate VC project of the SCARA robot cell. It is possible to develop Melfa Basic IV robot programs either with the Mitsubishi IDE (RT ToolBox2) or with the CIROS programming editor, which supports this native code too; they will be compiled into IRDATA programs for the use with the virtual robot controller of CIROS. The feasibility of the online coupling with the real Mitsubishi robot controller will be tested in the future work too.

The pending VC projects for both robot cells require similar CAD preparations as carried out for the transportation system. The decomposition of the gantry robot cell and the definition of components necessary for simulation according to the proposed procedure for mechatronic component model building (cf. fig. 5-9) have been already done, and the component model conceptualisation has been started. Likewise the CAD preparation following the well-proven procedure comprising simplification and component structuring into static units, moving and sensors has been successfully transferred to this new VC project. An overview is given in figure 10-1.

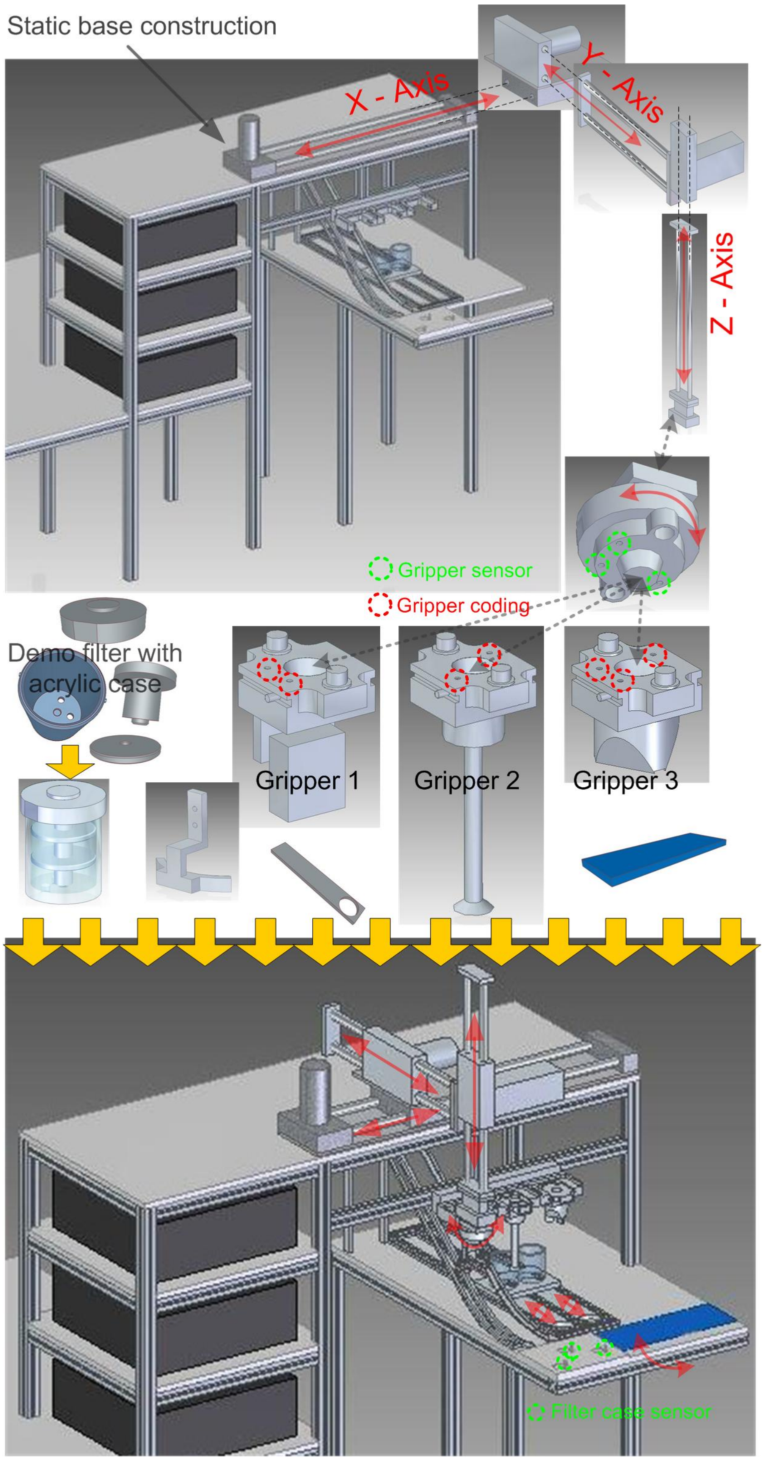


Fig. 10-1: Component definition and CAD preparation for gantry robot cell

The still outstanding Low-level Component Modelling in CIROS will presumably show the suitability of these component CAD models and thus allow for a VC.

After the final definition of operating processes for the SCARA robot cell and having added the missing peripheral devices a VC procedure will become possible for this robot cell too. The Mitsubishi

RH-20SDH8545 robot is included in the CIROS library, but all necessary devices of the surrounding working cell will have to pass CAD preparation and Low-level Component Modelling.

Generally, the implementation and execution of VC should be integrated in the education of students in automation and control engineering. The students should be aware of possibilities and drawbacks of VC and the possibly limited validity of simulation models and simulation results in this context. Due to its complexity, the RFAS is mostly used for more extensive projects in controller programming such as bachelor thesis projects, not in standard lab courses. Hence, a simulation model for a small educational sorting station has been developed in CIROS based on the procedures validated in this thesis, and the next step could be the testing of VC in different configurations, e.g. SIL with PLCSIM or HIL with CoDeSys based PLC. This model could be evaluated regarding its usability to introduce VC into lab courses at UASA Hannover.

Due to its industrial relevance, a future research on the integration of human workers (e.g. handwork place for loading/unloading the transport cars) into the VC of the RFAS should be undertaken. For this purpose, two different approaches are imaginable and worth to be evaluated:

- The integration of a virtual human (Rossmann et al., 2006, Schlette & Rossmann, 2009), which is provided as model with 30 degrees of freedom (DOF) by version 6 of CIROS (FESTO-Didactic, 2016). The research focus in this case would be the ability of this model to manipulate material (gripping etc.) or to act otherwise (e.g. pressing buttons) as part of the virtual manufacturing system during VC. The programmability of a virtual human model with 30 DOFs would be an interesting aspect for future work too.
- The application of semi- or fully-immersive VR (cf. subsection 2.7.2) seems to be promising for this use-case and is provided by CIROS (RIF, 2014). The research focus could be the investigation if immersive user interaction during VC at a handwork place is possible and reasonable.

To support the novel approach for exchangeable mechatronic component models outlined in subsection 5.5 it would be indispensable to produce 3D CAD models in COLLADA, thus a further evaluation of COLLADA based 3D CAD model design would become necessary. It turned out that COLLADA is not supported very well by the CAD tools available at the UASA Hannover. Only SolidWorks provided a “lab version” of an export filter for COLLADA (SolidWorks, 2013), which has been meanwhile discontinued. After additional installation of the necessary Nvidia PhysX software, it has been possible to export the CAD models of the transportation system using this filter (Fig. 10-2). All CAD components from the STEP library as well as the entire transportation system have been converted to COLLADA.

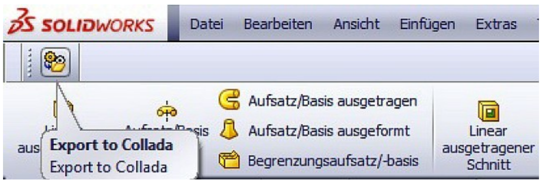


Fig. 10-2: SolidWorks menu bar with COLLADA exporter add-in

This export filter supports only COLLADA 1.4 without kinematics transfer possibility. A re-import of this COLLADA files to a CAD system at the UASA Hannover has not been possible, but tests with Blender and Google SketchUp (now Trimble SketchUp) have been successful (Fig. 10-3/10-4) and show the working COLLADA export.

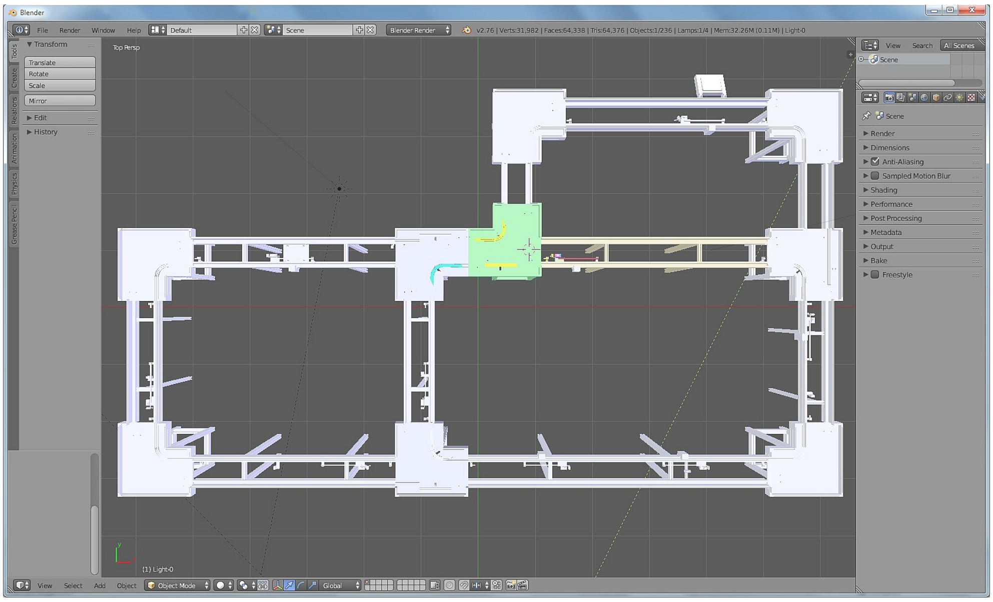
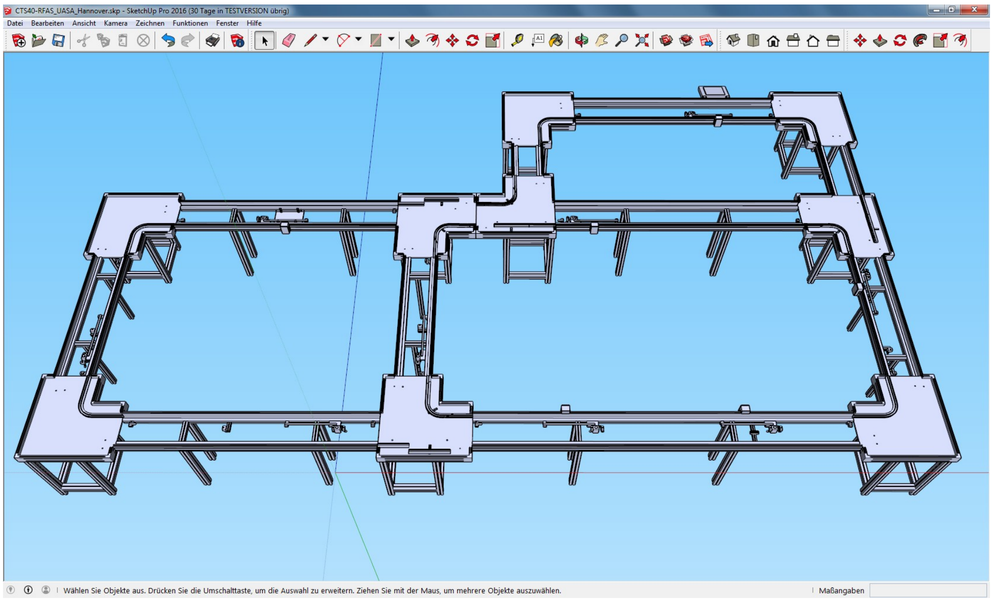


Fig. 10-3: COLLADA file (.dae) of entire transportation system imported in Blender

The 3D graphics suite Blender provides a lot of interesting animation and simulation features including kinematics and simulation of rigid bodies with collision detection but is generally not applied to the design or simulation of industrial manufacturing systems. Nevertheless, it is possible to use the integrated Blender game engine for such purpose as presented in (Lind & Skavhaug, 2011).


CTS40-RFAS_UASA_Hannover.skp - SketchUp Pro 2016 (30 Tage in TESTVERSION übrig)
Datei
Bearbeiten
Ansicht
Kamera
Zeichnen
Funktionen
Fenster
Hilfe
Maßangaben

Wählen Sie Objekte aus. Drücken Sie die Umschalttaste, um die Auswahl zu erweitern. Ziehen Sie mit der Maus, um mehrere Objekte auszuwählen.

Fig. 10-4: COLLADA file (.dae) of entire transportation system imported in SketchUp

The same restrictions apply to SketchUp which is neither commonly used for manufacturing systems, but for architectural use (e.g. building models for Google Earth). An exception has been presented by Dzinic and Yao (2013) by its use for preparation of COLLADA models to be imported to the simulation tool Experior.

Thus, a future task could be the further evaluation of these tools for the design of COLLADA models, but research regarding COLLADA support of more industrially accepted CAD tools or converter tool could be an interesting consideration for future work too.

The logics/behaviour model as part of the outlined novel mechatronic component models in subsection 5.5 facilitates PLCopen XML, thus another future task could be the further evaluation of logics/behaviour modelling supporting this approach. For this purpose, the model generation with an IDE supporting PLCopen XML, such as the typical PLC IDEs MultiProg (KW-Software), logi.CAD (logi.cals) or 3S-Codesys V3 ((PLCopen, 2011) could be tested.

Another possibility to generate PLCopen XML based logics/behaviour models could be the application of Simulink PLC Coder (Mathworks, 2015) or DEIF PLC Link (DEIF, 2015). Besides native code for several PLC IDEs, both tools can generate PLCopen XML.

References

3S - Smart Software Solutions (2014). *CODESYS – industrial IEC 61131-3 PLC programming*, Available: http://www.codesys.com/ [accessed December 7th, 2014]

ABB (2016). *Operating manual - RobotStudio 6.03.01,* Västerås, Sweden, p.590.

Abdul Kadir, A. and Xu, X. (2011). *Towards High-Fidelity Machining Simulation*, Journal of Manufacturing Systems, 30, 3, pp. 175-186.

Abdul Rahman, A. A. and Kernbaum, S. (2007). *Simulation Based Control System for a Manufacturing System*, Conference on Application and Design in Mechanical Engineering, Kangar, Perlis, Malaysia p. 7.

Addo-Tenkorang, R. (2011). *Concurrent Engineering (CE): A Review Literature Report*, Proceedings of the World Congress on Engineering and Computer Science 2011, San Francisco, USA, 2, p. 7.

Adolfsson, J., Ng, A., Olofsgard, P., Moore, P., Pu, J. and Wong, C.-B. (2002). *Design and simulation of component-based manufacturing machine systems*, Mechatronics, 12, No. 9-10, pp. 1239-1258.

Akesson, E. (2010). *Virtual Preparation of Advanced Production Systems,* Göteborg, Sweden, Chalmers University of Technology, Department of Signals and Systems - Division of Automatic Control, Automation and Mechatronics, Master's Thesis, p.72.

Allan, W. and Skibo, A. D. (2005). *A Practical Guide to Construction, Commissioning and Qualification Documentation - and its Critical Role in Achieving Compliance*, The Official Journal of ISPE, 25, 4, p. 8.

Altintas, Y., Brecher, C., Weck, M. and Witt, S. (2005). *Virtual machine tool*, CIRP Annals-Manufacturing Technology, 54, 2, pp. 115-138.

Andemeskel, F. (2013). *Model-Based Virtual Commissioning of a Manufacturing Cell Control System,* Stockholm, Sweden, Royal Institute of Technology (KTH), School of Industrial Engineering and Management, Master's thesis, p.89.

Andersson, C. and Runeson, P. (2002). *Verification and validation in industry - a qualitative survey on the state of practice*, 2002 International Symposium Empirical Software Engineering, pp. 37-47.

Angerbauer, R., Lewek, J. and Litto, M. (2004). *Funktionales Engineering - Baukasten für disziplinübergreifendes Projektieren*, iee, 49, 3, pp. 48-51.

Anton, O., Lercher, B. and Reinhart, G. (2002). *Modelling of Faults and Fault Recovery: An Essential Aspect of Mechatronic System Design*, Annals of 2002 Int'l CIRP Design Seminar, Hong Kong, p. 6.

AP242 (2016). *Introduction - STEP AP 242 On A Page*, Available: http://www.ap242.org/ [accessed July 31st, 2016]

Aquimo (2008). *Aquimo Projekt*, Available: http://www.hs-esslingen.de/de/forschung-transfer/institute-fuer-angewandte-forschung/institut-fuer-mechatronik/projekte/aquimo.html [accessed 10 May 2015]

Auinger, F., Vorderwinkler, M. and Buchtela, G. (1999). *Interface driven domain-independent modeling architecture for 'soft-commissioning' and 'reality in the loop'*, Proceedings of the 1999 Winter Simulation Conference (WSC '99), Phoenix, AZ, USA, 1, pp. 798-805.

Aurich, J. C., Biermann, D., Blum, H., Brecher, C., Carstensen, C., Denkena, B., Klocke, F., Kröger, M., Steinmann, P. and Weinert, K. (2009). *Modelling and simulation of process: machine interaction in grinding*, Production Engineering, 3, 1, pp. 111-120.

AutomationML (2014). *<AutomationML/>*, Available: http://www.automationml.org/ [accessed September 1st, 2014]

automotiveIT (2015). *CAD-Datenbrücke zwischen NX und CATIA*, Available: http://www.automotiveit.eu/cad-datenbruecke-zwischen-nx-und-catia/t-systems-subnet/id-0050119 [accessed July 31st, 2016]

Bacic, M. (2005). *On hardware-in-the-loop simulation*, Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on, pp. 3194-3198.

Bal, M. and Hashemipour, M. (2009). *Virtual factory approach for implementation of holonic control in industrial applications: A case study in die-casting industry*, Robotics and Computer-Integrated Manufacturing, 25, pp. 570–581.

Balci, O. (1990). *Guidelines for successful simulation studies*, Proceedings of the 1990 Winter Simulation Conference (WSC '90), pp. 25-32.

Baldwin, J. S., Rose-Anderssen, C. and Ridgway, K. (2011). *Linnaean and Cladistic Classification of Manufacturing Systems*, Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV2011), Montreal, Canada, pp. 29-34.

Ball, P. D., Roberts, S., Natalicchio, A. and Scorzafave, C. (2011). *Modelling production ramp-up of engineering products*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 225, 6, pp. 959-971.

Bani Younis, M. and Frey, G. (2003). *Formalization of existing PLC programs: A Survey*, CESA 2003, Lille (France), p. 6.

Banks, J. (1999). *Introduction to simulation*, Proceedings of the 1999 Winter Simulation Conference (WSC '99), 1, pp. 7-13 vol.11.

Banks, J., Carson II, J. S., Nelson, B. L. and Nicol, D. M. (2010). *Introduction to Simulation,* Discrete-Event System Simulation. 5th ed., Pearson Prentice Hall, p.640.

Banks, J. and Chwif, L. (2011). *Warnings about simulation*, Journal of Simulation, 5, 4, pp. 279-291.

Barreto, H. (2014). *Parallel Simulation For Concurrent Development Of Manufacturing Flow And Its Control System,* Stockholm, Sweden, KTH – The Royal Institute of Technology, ITM – School of Industrial Engineering and Management, IIP – Department of Production Engineering, Master's Thesis, p.65.

Beckers, R., Fröhlich, A. and Stjepandic, J. (2010). *Anwendung und Potenziale universeller Visualisierungsformate*, 9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, p. 13.

Beckhoff (2016a). *TwinCAT 3 – eXtended Automation Engineering (XAE)*, Available: https://www.beckhoff.com/english.asp?twincat/twincat-3-extended-automation-engineering.htm [accessed May 2nd, 2016]

Beckhoff (2016b). *TwinCAT 3 – eXtended Automation Runtime (XAR)*, Available: https://www.beckhoff.com/english.asp?twincat/twincat-3-extended-automation-runtime.htm [accessed May 2nd, 2016]

Bengtsson, K., Bergagard, P., Thorstensson, C., Lennartson, B., Åkesson, K., Yuan, C., Miremadi, S. and Falkman, P. (2012). *Sequence Planning Using Multiple and Coordinated Sequences of Operations*, IEEE Transactions on Automation Science and Engineering, 9, 2, pp. 308-319.

Bengtsson, K., Thorstensson, C., Lennartson, B., Åkesson, K., Yuan, C., Miremadi, S. and Falkman, P. (2010). *Relations identification and visualization for sequence planning and automation design*, 2010 IEEE International Conference on Automation Science and Engineering, pp. 841-848.

Berenbach, B. A., Spool, P. R. and Bitterle, D. (2003). *The Application of Modern Software Engineering Practices to Control Engineering*, Journal of Advanced Manufacturing Systems, 2, 1, p. 15.

Bergert, M., Kiefer, J., Höme, S. and Fedrowitz, C. (2009). *KUKA Report: Einsatz der Virtuellen Inbetriebnahme im automobilen Karosserierohbau – Ein Erfahrungsbericht*, p. 10.

Bernhardt, R., Schreck, G. and Willnow, C. (1995). *Realistic robot simulation*, COMPUTING & CONTROL ENGINEERING JOURNAL, 6, 4, pp. 174-176.

Bierschenk, S., Kuhlmann, T. & Ritter, A. (2005). *Stand der Digitalen Fabrik bei kleinen und mittelständischen Unternehmen*, Stuttgart, Fraunhofer-Institut für Produktionstechnik und Automatisierung -IPA-, p. 49.

Bischoff, R. (2007). *Anlaufmanagement - Schnittstelle zwischen Projekt und Serie,* Konstanz, Hochschule Konstanz Technik, Wirtschaft und Gestaltung, p.119.

Bishop, R. H. and Ramasubramanian, M. K. (2005). *What is Mechatronics?,* IN Bishop, R. H. (Ed.) Mechatronics: An Introduction. CRC Press, p.312.

Bockstette, H. (2013). *Untersuchung von Simulationssoftware zur 3D-Anlagensimulation,* Hannover, Germany, Hochschule Hannover, University of Applied Sciences and Arts, Faculty II, Mechanical Engineering, Automation and Control Engineering Group, Bachelor thesis, p.130.

Boehm, B. W. (1979a). *Guidelines for verifying and validating software requirements and design specifications*, Proceedings of the European Conference on Applied Information Technology of the International Federation for Information Processing (Euro IFIP'79), London, UK, pp. 711-719.

Boehm, B. W. (1979b). *Software engineering: R&D trends and defense needs,* IN Wegner, R. (Ed.) Research Directions in Software Technology. Cambridge, MA, USA, MIT Press, pp.44-86.

Bös, M. (2008). *Methoden der Digitalen Fabrikplanung - ein praxisorientierter Ansatz für KMU (Methods of the Digital Factory - an Application-oriented Approach for SME)*, 13. ASIM Fachtagung - Advances in Simulation for Production and Logistics Applications, Berlin, pp. 407-415.

Bosch-Rexroth (2014). *eShop - Transfersysteme TS 2plus*, Available: https://www.boschrexroth.com/irj/portal/anonymous/eShop?guest_user=anonymousDE&display=catalog&bridgeSelectedCatalog=MTN&bridgePageId=ts_2plus [accessed September 29th, 2014]

BOSCH (1992a). *Betriebsanleitung Transportwagen TW40 / TW60,* Waiblingen, Germany, Robert Bosch GmbH, p.33.

BOSCH (1992b). *Computergesteuertes Transfersystem CTS,* Waiblingen, Germany, Robert Bosch GmbH, p.39.

BOSCH (1992c). *CTSpro,* Waiblingen, Germany, Robert Bosch GmbH, p.11.

Bracht, U. and Reichert, J. (2010). *Digitale Fabrik – auch KMU sind aufgefordert künftig ihre Fabriken in 3D-CAD zu planen Ein Vorschlag zur modularen Auswahl und Einführung geeigneter Komponenten / The Digital Factory – SME are also urged to plan their factories with 3D-CAD*, Industrie Management, 2, pp. 65-68.

Brecher, C., Breitbach, T., Ecker, C. and Lohse, W. (2013a). *Environment sensing for the creation of work cell models*, Production Engineering, pp. 1-10.

Brecher, C., Esser, M. and Witt, S. (2009). *Interaction of manufacturing process and machine tool*, CIRP Annals - Manufacturing Technology, 58, 2, pp. 588-607.

Brecher, C., Nittinger, J. A. and Karlberger, A. (2013b). *Model-based Control of a Handling System with SysML*, Procedia Computer Science, 16, 0, pp. 197-205.

Burghardt, M. (2000). *Projektmanagement : Leitfaden für die Planung, Überwachung und Steuerung von Entwicklungsprojekten,* Erlangen, Publicis,

Bylinsky, G. (1994). *The Digital Factory*, Fortune international, 130, 10, pp. 50-59.

Byrne, J., Liston, P., Ferreira, D. C. e. and Byrne, P. (2015). *Cloud based Data Capture and Representation for Simulation in Small and Medium Enterprises*, Proceedings of the 2015 Winter Simulation Conference (WSC '15), Huntington Beach, CA, USA, pp. 2195-2206.

Cadenas (2014). *2D & 3D CAD MODELS - Manufacturer catalogs*, Available: http://b2b.partcommunity.com/community/ [accessed September 29th, 2014]

Caputo, F., Di Gironimo, G. and Marzano, A. (2006). *A structured approach to simulate manufacturing systems in virtual environment*, XVIII Congreso International de Ingegneria Grafica, Barcelona, Spain.

Cardoso, L. d. D., Rangel, J. J. d. A. and Bastos, P. J. T. (2013). *Discrete event simulation for integrated design in the production and commissioning of manufacturing systems*, Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World, Washington, D.C., pp. 2544-2552

Carlsson, H., Danielsson, F. and Lennartson, B. (2008). *General Time Synchronisation Method for PLC Programs*, Proceedings of the 17th IFAC World Congress, Seoul, Korea.

Carlsson, H., Svensson, B., Danielsson, F. and Lennartson, B. (2012). *Methods for Reliable Simulation-Based PLC Code Verification*, Industrial Informatics, IEEE Transactions on, 8, 2, pp. 267-278.

Carson II, J. S. (1989). *Verification and validation: a consultant's perspective*, Proceedings of the 21st Winter simulation conference (WSC '89), Washington, D.C., USA, pp. 552-558.

Choi, S., Jung, K. and Noh, S. D. (2015). *Virtual reality applications in manufacturing industries: Past research, present findings, and future directions*, Concurrent Engineering, 23, 1, pp. 40-63.

Chwif, L., Barretto, M. R. P. and Paul, R. J. (2000). *On simulation model complexity*, Proceedings of the 2000 Winter Simulation Conference, 1, pp. 449-455 vol.441.

Clark, W. (1923). *The Gantt chart - a working tool of management,* New York, USA, Ronald Press, Available: https://archive.org/details/ganttchartworkin00claruoft, p.190.

Cochran, D., Zhao, Z. and Ng, Q. (2001). *The Role of Physical Simulation in the Re-Design of Existing Manufacturing Systems*, CIRP Design Seminar, Stockholm, Sweden, p. 6.

Cruz, F. d., Nunes, L. E. N. d. P., Correa, V. A. and Grandinetti, F. J. (2014a). *Comissionamento Virtual: Ferramenta de Validação de Programas de Controle de Sequencia em Sistemas Automatizados de Manufatura*, XI Simpósio de Excelência em Gestão e Tecnologia (SEGeT 2014), Campo de Aviação, Resende, Brazil, p. 13.

Cruz, F. d., Nunes, L. E. N. d. P., Correa, V. A. and Grandinetti, F. J. (2014b). *COMISSIONAMENTO VIRTUAL: FERRAMENTA DE VALIDAÇÃO DE PROGRAMAS DE SEQUENCIAS AUTOMATIZADAS DE MANUFATURA*, III Congresso Internacional de Ciência, Tecnologia e Desenvolvimento - CIÊNCIA E TECNOLOGIA PARA O DESENVOLVIMENTO SOCIAL, Universidade de Taubaté (UNITAU), Brazil, p. 9.

Danielsson, F., Moore, P. and Eriksson, P. (2003). *Validation, off-line programming and optimisation of industrial control logic*, Mechatronics, 13, 6, pp. 571-585.

Dassault Systemes (2014). *Delmia - Validation of NC Machine Tools*, Available: http://www.3ds.com/products-services/3dexperience/machining/nc-machine-code-validation/ [accessed August 28, 2014]

Davidson, J. and Sennö, T. (2005). *Interactive Control of a Virtual Machine,* Lund, Sweden, Lund University, Department of Industrial Electrical Engineering and Automation, Master's Thesis, p.102.

DEIF (2015). *PLC Link Code generation - The efficient way to generate flawless IEC 61131 compliant code*, Available: http://www.deifwindpower.com/technology/plc-link-code-generation [accessed February 17th, 2015]

Demo3D (2016). *Connectivity - Emulate3D Connects to the Controls World*, Available: http://www.demo3d.com/Connectivity/ [accessed May 1st, 2016]

Denkena, B., Ammermann, C. and Hoppe, P. (2008). *Ramp-Up/2 – Verkürzung von Fertigungsanläufen - Ramp-Up/2 - Shortening of ramp-up-time*, wt Werkstattstechnik online, 98, 3, pp. 143-148.

Denkena, B. and Hollmann, F. (2012). *Process Machine Interactions: Predicition and Manipulation of Interactions Between Manufacturing Processes and Machine Tool Structures*, Springer Science & Business Media, p.518

Dépincé, P., Chablat, D. and Woelk, P.-O. (2004). *Virtual Manufacturing : Tools for improving Design and Production*, 14th CIRP International Design Seminar, Cairo, Egypt.

Digital-Engineering (2016). *Industrielle Physiksimulationen mit der VR-Brille Oculus-Rift*, Available: http://www.digital-engineering-magazin.de/industrielle-physiksimulationen-mit-der-vr-brille-oculus-rift# [accessed April 19th, 2016]

DIN 32541, (1977). *Betreiben von Maschinen und vergleichbaren technischen Arbeitsmitteln; Begriffe für Tätigkeiten - Management of machines and similar technical equipment; Terminology associated with activities.*

DIN 19246, (1991). *Messen, Steuern, Regeln; Abwicklung von Projekten; Begriffe - Measurement and control; procedure of project; terms and definition.*

DIN EN 62381, (2013). *Automatisierungssysteme in der vefahrenstechnischen Industrie - Werksabnahme (FAT), Abnahme der installierten Anlage (SAT) und Integrationstest (SIT) - Automation systems in the process industry - Factory acceptance test (FAT), site acceptance test (SAT) and site integration test (SIT), Also British Standard BS EN 62381 (2012-05-31).*

DIN EN 62264-1, (2014). *Integration von Unternehmensführungs- und Leitsystemen – Teil 1: Modelle und Terminologie, IEC 62264-1:2013); Deutsche Fassung EN 62264-1:2013 / Enterprise-control system integration – Part 1: Models and terminology (IEC 62264-1:2013); German version EN 62264-1:2013*, p. 84.

Dominka, S., Schiller, F. and Kain, S. (2007). *Hybrid commissioning: from hardware-in-the-loop simulation to real production plants*, Proceedings of the 18th IASTED International Conference: modelling and simulation, Montreal, Canada, pp. 544-549.

Dougall, D. J. (1998). *Applications and benefits of real-time I / 0 simulation for PLC and PC control systems*, ISA Transactions, 36, 4, pp. 305-311.

Drath, R., Weber, P. and Mauser, N. (2008a). *An evolutionary approach for the industrial introduction of virtual commissioning*, IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008), Hamburg, Germany, pp. 5-8.

Drath, R., Weber, P. and Mauser, N. (2008b). *Virtuelle Inbetriebnahme – ein evolutionäres Konzept für die praktische Einführung*, Automation 2008, Baden-Baden, VDI-Berichte 2032, pp. 73-76.

Dubey, A. (2011). *Evaluating software engineering methods in the context of automation applications*, 2011 9th IEEE International Conference on Industrial Informatics (INDIN '11), pp. 585-590.

Dumitrascu, A., Nae, L. and Predincea, N. (2014). *VIRTUAL COMMISSIONING AS A FINAL STEP IN DIGITAL VALIDATION OF THE ROBOTIC MANUFACTURING SYSTEMS*, Proceedings in Manufacturing Systems, 9, 4, pp. 215-220.

Dzinic, J. and Yao, C. (2013). *Simulation-based verification of PLC programs,* Gothenburg, Sweden, Chalmers University of Technology, Department of Signals and Systems, Master's thesis, p.43.

EC - The European Parliament and the Council of the European Union Official Journal of the European Union (2006). *DIRECTIVE 2006/42/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 17 May 2006 on machinery, and amending Directive 95/16/EC (recast).*

European Commission (2015a). *Facts and Figures*, Research & Innovation, SME Techweb, SME Update: Issue 15, Available: http://ec.europa.eu/research/sme-techweb/newsletter/issue15/facts-figures_en.html [accessed 11/22/2015]

European Commission (2015b). *What is an SME?*, Growth, Entrepreneurship and SMEs, Available: http://ec.europa.eu/growth/smes/business-friendly-environment/sme-definition/index_en.htm [accessed 11/22/2015]

Eckes, R. and Wagner, R. (2006). *Einsatz von Augmented Reality im Ramp-Up Prozess von automatisierten Fertigungssystemen*, HNI-Verlagsschriftenreihe, 188, p. 13.

ElMaraghy, H. A., AlGeddawy, T., Azab, A. and ElMaraghy, W. (2011). *Change in Manufacturing - Research and Industrial Challenges*, Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV2011), Montreal, Canada, pp. 2-9.

Elysium (2015). *Interactive geometry verification and healing for multi-CAD data exchange, geometry simplification for CAE, plus tools for Rapid Prototyping and Reverse Engineering*, Available: http://elysiuminc.com/products/caddoctor/ [accessed February 10th, 2015]

Erlandsson, T. and Rahaman, M. M. (2013). *Testing and verifying PLC code with a virtual model of Tetra Pak Filling Machine,* Gothenburg, Sweden, Chalmers University of Technology, Department of Signals and systems, Master's Thesis, p.37.

Estévez, E., Marcos, M., Lüder, A. and Hundt, L. (2010). *PLCopen for achieving interoperability between development phases*, Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on, pp. 1-8.

Eversheim, W., Koerth, D. and Gentzcke, J. (1990). *Inbetriebnahme komplexer Maschinen und Anlagen : Strategien und Praxisbeispiele zur Rationalisierung in der Einzel- und Kleinserienproduktion,* Düsseldorf, Verein Deutscher Ingenieure, VDI-Gesellschaft Produktionstechnik (ADB). Walter Eversheim (Ed.), p.142.

Fehsenfeld, B. (2003). *Engineering: Systematische Entwicklungsprozesse bringen neuen Schwung in die Elektronik - Integration Maschinenbau profitiert von modularer Steuerungssoftware*, VDI Nachrichten, 04.7.2003, VDI, Frankfurt/Main

FESTO Didactic (2016). *CIROS® Studio – Creating virtual learning environments*, Available: http://www.festo-didactic.com/int-en/learning-systems/software-e-learning/ciros/ciros-studio-creating-virtual-learning-environments.htm [accessed August 14th, 2016]

FESTO Didactic (2015a). *CIROS® Automation Suite*, Available: http://www.festo-didactic.com/int-en/learning-systems/software-e-learning/ciros-automation-suite [accessed January 13th, 2015]

FESTO Didactic (2015b). *Robotino® – For research and education: Premium Edition and Basic Edition*, Available: http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/robotino-for-research-and-education-premium-edition-and-basic-edition.htm [accessed January 5th, 2015]

FESTO Didactic (2015c). *Robotino® XXT*, Available: http://www.festo-didactic.com/int-en/services/robotino/robotino-xxt/ [accessed January 5th, 2015]

Föderal (2001-2004). *Föderal Projekt*, Available: http://www.produktionsforschung.de/PFT/verbundprojekte/vp/index.htm?VP_ID=522 [accessed March 5th, 2014]

Franklin, B. (1748). *Advice to a Young Tradesman, Written by an Old One,* IN Houston, A. (Ed.) Franklin: The Autobiography and Other Writings on Politics, Economics, and Virtue. Cambridge University Press 2004, pp.200-202.

Freund, E., Ludemann-Ravit, B., Stern, O. and Koch, T. (2001a). *Creating the architecture of a translator framework for robot programming languages*, International Conference on Robotics & Automation (ICRA 2001), Seoul, South Korea, 1, pp. 187-192.

Freund, E. and Pensky, D. H. (2002). *COSIMIR Factory: Extending the use of manufacturing simulations*, International Conference on Robotics & Automation, Washington, DC, United States, 3, pp. 2805-2810.

Freund, E. and Rossmann, J. (1995). *Systems approach to robotics and automation*, Proceedings of the 1995 IEEE International Conference on Robotics and Automation, Nagoya vol.1, pp. 3-14.

Freund, E., Rossmann, J., Uthoff, J. and van der Valk, U. (1994). *Towards realistic simulation of robotic workcells*, IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS '94), 'Advanced Robotic Systems and the Real World', Munich, Germany, 1, pp. 39-46.

Freund, E., Schluse, M. and Rossmann, J. (2001b). *State oriented modeling as enabling technology for projective virtual reality*, International Conference on Intelligent Robots and Systems, Maui, HI, 4, pp. 1842-1847.

Freund, E., Uthoff, J., Hypki, A. and van der Valk, U. (1993). *COSIMIR und PCROB: Integration von Zellensimulation und Robotersteuerung auf PCs*, VDI/VDE-GMA-Fachtagung Intelligente Steuerung und Regelung von Robotern, Langen b. Frankfurt/M., Germany, VDI Berichte 1094, pp. 823-834.

Frey, G. (2002). *Formal methods in PLC Control demonstrated at a flexible manufacturing line*, American Control Conference, pp. 501-508.

Friedewald, A., Lödding, H., Lukas, U. F. v., Mesing, B., Roth, M., Schleusener, S. and Titov, F. (2011). *Benchmark neutraler Formate für den prozessübergreifenden Datenaustausch im Schiffbau*, Fraunhofer IGD, p.29.

Fröhlich, A. (2013). *Whitepaper: 3D Formats in the Field of Engineering - a Comparison,* Darmstadt, Germany, PROSTEP AG, p.24.

Gantt, H. L. (1903). *A graphical daily balance in manufacture*, Transactions of the American Society of Mechanical Engineers (ASME Transactions), Ney York City, USA, 24, Available: https://babel.hathitrust.org/cgi/pt?id=mdp.39015023119541;view=1up;seq=1358, pp. 1322–1336.

Gausemeier, J., Grafe, M. and Ebbesmeyer, P. (2000). *Nutzenpotenziale von Virtual Reality in der Fabrik- und Anlagenplanung / Benefits of virtual reality for factory and plant design*, wt Werkstattstechnik online, 90, 7/8, pp. 282-286.

Glas, J. (1993). *Standardisierter Aufbau Anwendungsspezifischer Zellenrechnersoftware,* Munich, Germany, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb), Dissertation, p.139.

Greifeneder, J. and Gohr, K. (2014). *Smart FAT – Zur Sicherheit Automatisiert / Smart FAT – Automated execution for an increased level of safety*, Automation 2014, Baden-Baden, Germany, VDI-Berichte 2231, pp. 473-487.

Grimm, B. (2011). *AutomationML*, Presentation, IEC TC65: Industrial Automation Forum, Seoul, South Korea, May 18th, 2011, p. 30.

Grimm, B. (2012). *Virtuelle Inbetriebnahme von Produktionsanlagen*, atp edition – Automatisierungstechnische Praxis, 54, 4, pp. 28-33.

Gu, F., Harrison, W. S., Tilbury, D. M. and Chengyin, Y. (2007). *Hardware-In-The-Loop for Manufacturing Automation Control: Current Status and Identified Needs*, IEEE International Conference on Automation Science and Engineering 2007 (CASE '07) Scottsdale, AZ, USA, pp. 1105-1110.

Guerrero, L. V., López, V. V. and Mejía, J. E. (2014). *Virtual Commissioning with Process Simulation (Tecnomatix)*, Computer-Aided Design and Applications, 11, sup1, pp. 11-19.

Haenisch, J., Kroszynski, U., Ludwig, A. and Sørensen, T. (1996). *Specification of a STEP Based Reference Model for Exchange of Robotics Models : Geometry, Kinematics, Dynamics, Control, and Robotics Specific Data*, Forschungszentrum Karlsruhe, p.493.

Hametner, R., Kormann, B., Vogel-Heuser, B., Winkler, D. and Zoitl, A. (2011). *Test case generation approach for industrial automation systems*, Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, pp. 57-62.

Hanselmann, H. (1996). *Hardware-in-the-loop simulation testing and its integration into a CACSD toolset*, Computer-Aided Control System Design, 1996., Proceedings of the 1996 IEEE International Symposium on, pp. 152-156.

Haq, I., Monfared, R., Harrison, R., Lee, L. and West, A. (2010). *A new vision for the automation systems engineering for automotive powertrain assembly*, International Journal of Computer Integrated Manufacturing, 23, 4, pp. 308-324.

Harashima, F., Tomizuka, M. and Fukuda, T. (1996). *Mechatronics -"What Is It, Why, and How?" An Editorial* IEEE/ASME Transactions on Mechatronics, 1, 1, pp. 1-4.

Harrison, W. S. (2011). *Virtual Fusion: The Integration and Analysis of Simulation and Real Processes for Manufacturing Process Deployment*, University of Michigan, PhD thesis, p.111.

Harrison, W. S. and Proctor, F. (2015). *Virtual Fusion: State of the Art in Component Simulation/Emulation for Manufacturing*, Procedia Manufacturing, 1, pp. 110-121.

Harrison, W. S. and Tilbury, D. (2008). *Virtual Fusion: Hybrid Process Simulation and Emulation-in-the-Loop*, 9th Biennial Conference on Engineering Systems Design and Analysis (ASME 2008), Volume 1: Advanced Energy Systems; Advanced and Digital Manufacturing; Advanced Materials; Aerospace, Haifa, Israel, pp. 263-270.

Hästbacka, D., Vepsäläinen, T. and Kuikka, S. (2011). *Model-driven development of industrial process control applications*, Journal of Systems and Software, 84, 7, pp. 1100-1113.

Hedlind, M., Klein, L., Li, Y. and Kjellberg, T. (2011). *Kinematic structure representation of products and manufacturing resources*, Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology, Athens, Greece.

Himmler, F. and Amberg, M. (2013). *Die Digitale Fabrik - eine Literaturanalyse*, 11th International Conference on Wirtschaftsinformatik, Leipzig, Germany, pp. 165-179.

Hincapié, M., Ramírez, M. d. J., Valenzuela, A. and Valdez, J. A. (2014). *Mixing real and virtual components in automated manufacturing systems using PLM tools*, International Journal on Interactive Design and Manufacturing (IJIDeM), 8, 3, pp. 209-230.

HMS Industrial Networks GmbH (2016). *Normenwerk industrieller Netze*, Available: http://www.feldbusse.de/Normung/normung.shtml [accessed May 5th, 2016]

Hoffmann, P. (2013). *Modellierung für die Virtuelle Inbetriebnahmee (VIBN) - Verhaltensmodellierung Transportwagen*, Presentation, AutomationML-Workshop "Use-Cases for FBD", June 19th, 2013, Phoenix Contact, Bad Pyrmont, Germany

Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2009). *Analyse und Konzepte zur vereinfachten Virtuellen Inbetriebnahme von Fertigungssystemen - Analysis and concepts for simplified virtual commissioning of manufacturing systems* Automation 2009, Baden-Baden, Germany, VDI-Berichte 2067, pp. 481-484.

Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2010). *Virtual Commissioning of Manufacturing Systems - A Review and new Approaches for Simplification*, Proceedings of the 24th European Conference on Modelling and Simulation (ECMS 2010), Kuala Lumpur, Malaysia, pp. 175-181.

Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2012). *Research on simplified modelling strategy for virtual commissioning*, 24th European Modeling and Simulation Symposium, EMSS 2012, Vienna, Austria, pp. 293-302.

Hollander, A. and Sappei, S. (2011). *Virtual preparation of Tetra Pak Filling Machine,* Göteborg, Sweden, CHALMERS UNIVERSITY OF TECHNOLOGY, Department of Signals and Systems; Division of Automatic Control, Automation and Mechatronics, Master's Thesis, p.50.

Hong, X. and Jianhua, W. (2006). *Using standard components in automation industry: A study on OPC Specification*, Computer Standards & Interfaces, 28, 4, pp. 386-395.

Hönle, R. (2014). *Speicherprogrammierbare Steuerungen,* IN Hering, E., Bressler, K. & Gutekunst, J. (Eds.) Elektronik für Ingenieure und Naturwissenschaftler. Springer Berlin Heidelberg, p.833.

Hossain, M. and Semere, D. T. (2013). *Virtual Control System Development Platform with the Application of PLC Device*, Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS 20013), Hong Kong, China, p. 6.

Hoyer, M., Schumann, R., Hoffmann, P. and Premier, G. C. (2008). *Virtuelle Inbetriebnahme mit Model*^{CAT}*- Vom Prototypen zum industriellen Einsatz / Virtual Start-Up with Model*^{CAT}*- From Prototypical Realisation to Industrial Implementation*, Automation 2008, Baden-Baden, Germany, VDI-Berichte 2032, pp. 203-206.

Hoyer, M., Schumann, R. and Premier, G. C. (2006). *Industrial CACSD for the plant design process*, 2006 IEEE, Computer Aided Control System Design, International Conference on Control Applications, International Symposium on Intelligent Control, pp. 3030-3035.

ICAM (2014). *ICAM NC Manufacturing Suite*, Available: http://www.icam.com/html/products/product.php [accessed August 28, 2014]

IEC 61131-3, (2003). *Programmable Controllers Part 3: Programming Languages. Also British Standard BS EN 61131-3:2003.*

IEC 62714-1, ed.1.0 (2014). *Engineering data exchange format for use in industrial automation systems engineering - Automation markup language - Part 1: Architecture and general requirements.*

IEEE (2012). *IEEE Standard for System and Software Verification and Validation, IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004)*, p. 223.

INCOSE (2011). *Systems Engineering Handbook V. 3.2.2 - A Guide for System Life Cycle Processes and Activities,* San Diego, CA, USA, International Council on Systems Engineering (INCOSE), p.386.

Isermann, R., Schaffnit, J. and Sinsel, S. (1999). *Hardware-in-the-loop simulation for the design and testing of engine-control systems*, Control Engineering Practice, 7, 5, pp. 643-653.

ISO (1994). *ISO 10303-1:1994 Industrial automation systems and integration -- Product data representation and exchange -- Part 1: Overview and fundamental principles*, p. 17.

ISO (2010). *ISO 10303-214:2010 Industrial automation systems and integration -- Product data representation and exchange -- Part 214: Application protocol: Core data for automotive mechanical design processes.*

ISO (2011). *ISO 10303-203:2011 Industrial automation systems and integration -- Product data representation and exchange -- Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies.*

ISO (2014). *ISO 10303-242:2014 Industrial automation systems and integration -- Product data representation and exchange -- Part 242: Application protocol: Managed model-based 3D engineering*, p. 11.

ITI-Transcendata (2015). *CAD Translation, Healing & Repair*, Available: http://www.transcendata.com/products/cadfix/index.htm [accessed February 10th, 2015]

Jae Ick, L., Sung Wook, C. and Soon Ju, K. (2002). *Virtual prototyping of PLC-based embedded system using object model of target and behavior model by converting RLL-to-statechart directly*, J. Syst. Archit., 48, 1-3, pp. 17-35.

Jain, A., Vera, D. A. and Harrison, R. (2010). *Virtual Commissioning of Modular Automation Systems*, 10th IFAC Workshop on Intelligent Manufacturing Systems (2010), Lisbon, Portugal.

Jo, J. Y., Kim, Y., Podgurski, A. and Newman, W. S. (1997). *Virtual testing of agile manufacturing software using 3D graphical simulation*, Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on, 2, pp. 1223-1228 vol.1222.

Johnstone, M., Creighton, D. and Nahavandi, S. (2007). *Enabling industrial scale simulation / emulation models*, Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come, Washington D.C., USA, pp. 1028-1034.

Kabitzsch, K., Naake, J., Roeder, A. and Vasyutynskyy, V. (2008). *Rapid migration and commissioning of industrial equipment*, Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, Hamburg, pp. 725-728.

Kain, S., Dominka, S., Merz, M. and Schiller, F. (2009). *Reuse of HiL simulation models in the operation phase of production plants*, Industrial Technology, 2009. ICIT 2009. IEEE International Conference on, pp. 1-6.

Kanai, S., Iyoda, D., Endo, Y., Sakamoto, H. and Kanatani, N. (2012). *Appearance preserving simplification of 3D CAD model with large-scale assembly structures*, International Journal on Interactive Design and Manufacturing, pp. 1-16.

Kiefer, J. (2007). *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau,* Saarbrücken, Universität des Saarlandes, H. Bley und C. Weber, Lehrstuhl für Fertigungstechnik/CAM, Dissertation, p.179.

Kiefer, J., Baer, T. and Bley, H. (2006). *Mechatronic-oriented Engineering of Manufacturing Systems - Taking the Example of the Body Shop*, PROCEEDINGS OF LCE2006, pp. 681-686.

Kiefer, J., Bergert, M. and Rossdeutscher, M. (2011). *Mechatronic Objects in Production Engineering - A Key Enabler in Automotive Industry*, automatisieren!, 13, pp. 15-30.

Kim, Y. S., Shin, K. Y., Lee, J. H., Lee, S. S., Kim, K. S., Kang, K. C. and Yang, J. S. (2013). *Application of virtual commissioning technology in a steel making industry*, 13th International Conference on Control, Automation and Systems (ICCAS 2013), Gwangju, Korea, pp. 1718-1720.

Kjellberg, T., von Euler-Chelpin, A., Hedlind, M., Lundgren, M., Sivard, G. and Chen, D. (2009). *The machine tool model—A core part of the digital factory*, CIRP Annals - Manufacturing Technology, 58, 1, pp. 425-428.

Klemm, P. and Korajda, I. (2003). *Component-Based Engineering for Machine Tool and Plant Builders*, Modern Trends in Manufacturing. Centre for Advanced Manufacturing Technologies., Wroclaw, pp. 199-205.

Klingstam, P. and Gullander, P. (1999). *Overview of simulation tools for computer-aided production engineering*, Computers in Industry, 38, pp. 173-186.

Ko, M., Chang, D. S. and Park, S. C. (2012). *Virtual Commissioning For PLC Simulation*, 26th Annual European Simulation and Modelling Conference (ESM'2012), Essen, Germany.

Köbler, J. & Pleuler, M. (2011). *Digitale Fabrikplanung bei KMUs*, IAF - Beiträge aus Forschung und Technik, Offenburg, Germany, Institut für Angewandte Forschung (IAF), Hochschule Offenburg, pp. 105-107.

Kong, X., Ahmad, B., Harrison, R., Park, Y. and Lee, L. J. (2012). *Direct deployment of component-based automation systems*, Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on, pp. 1-4.

Konnov, A. (2007). *Zuverlässigkeitsberechnung und vorbeugende Wartung von komplexen technischen Systemen mittels modifizierter Markov-Methode,* Karlsruhe, Germany, Universität Karlsruhe (TH), Fakultät Informatik, Dissertation, p.155.

Korajda, I., Seyfarth, M. and Pritschow, G. (2004). *Disziplinübergreifende Baukastensysteme*, wt Werkstattstechnik online, 94, 5, pp. 215-219.

Körner, S. (1999). *A Structured Approach to Identification Techniques for the Analysis of Industrial Processes,* Pontypridd, University of Glamorgan, School of Technology, PhD Thesis,

Krause, H. (2007). *Virtual Commissioning of a large LNG plant with the DCS "800xA" by ABB*, 6th EUROSIM Congress on Modelling and Simulation, Ljubljana p. 7.

Kuehn (2006a). *DIGITAL FACTORY – INTEGRATION OF SIMULATION ENHANCING THE PRODUCT AND PRODUCTION PROCESS TOWARDS OPERATIVE CONTROL AND OPTIMISATION*, Int. Journal of Simulation, 07, 07, pp. 27-39.

Kuehn, W. (2006b). *DIGTAL FACTORY - INTEGRATION OF SIMULATION FROM PRODUCT AND PRODUCTION PLANNING TOWARDS OPERATIVE CONTROL*, Proceedings 20th European Conference on Modelling and Simulation (ECMS 2006).

Kuhlenkötter, B., Schyja, A., Hypki, A. and Miegel, V. (2010). *Robot Workcell Simulation with AutomationML Support - An Element of the CAx-Tool Chain in Industrial Automation*, Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), Munich, Germany, pp. 1076-1082.

KUKA (2014). *KUKA youBot store*, Available: http://www.youbot-store.com/home [accessed March 14th, 2014]

Kunze, S. (2012). *Vergleich ausgewählter Datenaustauschstrategien im Ingenieurwesen,* Magdeburg, Germany, Otto-von-Guericke-Universität Magdeburg, Fak. Informatik, Master's Thesis, p.115.

Kwak, J.-G., Park, S. and Chang, M. (2010). *Geometric data simplification for a virtual factory*, The International Journal of Advanced Manufacturing Technology, 50, 1, pp. 409-418.

Lämkull, D., Hanson, L. and Roland, Ö. (2009). *A comparative study of digital human modelling simulation results and their outcomes in reality: A case study within manual assembly of automobiles*, International Journal of Industrial Ergonomics, 39, 2, pp. 428-441.

Lampérière-Couffin, S., Rossi, O., Roussel, J.-M. and Lesage, J.-J. (1999). *Formal validation of PLC programs: a survey*, European Control Conference, p. 6.

Law, A. M. (2009). *How to build valid and credible simulation models*, Proceedings of the 2009 Winter Simulation Conference (WSC '09), Austin, TX, USA, pp. 24-33.

LeBaron, T. and Jacobsen, C. (2007). *The simulation power of Automod*, Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come (WSC '07), Washington D.C., pp. 210-218.

Lechler, A., Kircher, C. and Verl, A. (2008). *Simulationsgestütztes mechatronisches Engineering - Entwurf und Inbetriebnahme mechatronischer Module und flexibler Produktionssysteme*, wt Werkstattstechnik online, 98, 5, p. 377.383.

Lee, C. G. and Park, S. C. (2014). *Survey on the virtual commissioning of manufacturing systems*, Journal of Computational Design and Engineering, 1, 3, pp. 213-222.

Lee, S. M., Harrison, R., West, A. A. and Ong, M. H. (2007). *A component-based approach to the design and implementation of assembly automation system*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 221, 5, pp. 763-773.

Lemmer, K., Ober, B. and Schnieder, E. (1995). *Model-based programming and diagnosis for programmable logical controllers*, Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on, 5, pp. 4474-4479.

Li, K. (2011a). *3D modelling and simulation of a production line with CIROS,* Tampere, Finland, Tampere University of Technology, Faculty of Automation, Mechanical and Materials Eng., Masters Thesis, p.78.

Li, Y. (2011b). *Implementation and evaluation of kinematic mechanism modeling based on ISO 10303 STEP,* Stockholm, Sweden, Royal Institute of Technology, Department of Production Engineering, Master's Thesis, p.45.

Li, Y., Hedlind, M. and Kjellberg, T. (2011). *Implementation of kinematic mechanism data exchange based on STEP*, Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology, Athens, Greece, pp. 152-159.

Li, Y., Hedlind, M., Kjellberg, T. and Sivard, G. (2015). *System integration for kinematic data exchange*, International Journal of Computer Integrated Manufacturing, 28, 1, pp. 87-97.

Lind, M. and Skavhaug, A. (2011). *Using the blender game engine for real-time emulation of production devices*, International Journal of Production Research, 50, 22, pp. 6219-6235.

Litto, M. (2006). *Ist Mechatronik beherrschbar?*, Intelligenter Produzieren, 2, pp. 13-15.

Litz, L., Suhm, C. and Schumann, R. (1998). *Computer Aided Engineering (CAE) in der Automatisierungstechnik*, atp - Automatisierungstechnische Praxis, 1998, 06, pp. 54-64.

Lobo, E., Fertuzinhos, J., Silva, J. P. M. A. and Machado, J. (2013). *Obtaining Plant Models for Formal Verification Tasks from 3D CAD Models: Which is the Best Approach?* , Advanced Materials Research, 630, pp. 283-290

Lobov, A., Lastra, J. L. M. and Tuokko, R. (2005). *On controller and plant modeling for model-based formal verification*, Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on, 1, pp. 8 pp.-128.

Lüder, A., Estévez, E., Hundt, L. and Marcos, M. (2010a). *Automatic transformation of logic models within engineering of embedded mechatronical units*, The International Journal of Advanced Manufacturing Technology, 54, 9, pp. 1077-1089.

Lüder, A., Foehr, L. H. M., Wagner, T., Zaddach, J. J. and Holm, T. (2010b). *Manufacturing system engineering with mechatronical units*, Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on, pp. 1-8.

Lüder, A., Hundt, L. and Keibel, A. (2010c). *Description of manufacturing processes using AutomationML*, Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on, pp. 1-8.

Machado, J., Denis, B. and Lesage, J.-J. (2006). *A generic approach to build plant models for DES verification purposes*, Discrete Event Systems, 2006 8th International Workshop on, pp. 407-412.

Machado, J. and Seabra, E. (2013). *HiL simulation workbench for testing and validating PLC programs*, 11th IEEE International Conference on Industrial Informatics (INDIN 2013) pp. 230-235.

Maclay, D. (1997). *Simulation gets into the loop*, IEE Review, 43, 3, pp. 109-112.

Maga, C., Jazdi, N. and Göhner, P. (2011). *Reusable Models in Industrial Automation: Experiences in Defining Appropriate Levels of Granularity*, Proceedings of the 18th IFAC World Congress, Università Cattolica del Sacro Cuore, Milano, Italy, pp. 9145-9150.

Magnus, S., Süß, S., Strahilov, A., Gulan, S. and Krause, J. (2015). *Testautomatisierung in der virtuellen Inbetriebnahme*, Automation 2015, Baden-Baden, Germany, pp. 121-134.

Mai, S. and Yi, M. J. (2010). *A study on OPC specifications: Perspective and challenges*, Strategic Technology (IFOST), 2010 International Forum on, pp. 193-197.

Makris, S., Michalos, G. and Chryssolouris, G. (2012). *Virtual Commissioning of an Assembly Cell with Cooperating Robots*, Advances in Decision Sciences, 2012, p. 11.

Mandel, S., Bar, T. and Fay, A. (2008). *Concept for proactive ramp-up validation of body-in-white lines*, Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, pp. 693-696.

Maropoulos, P. G. and Ceglarek, D. (2010). *Design verification and validation in product lifecycle*, CIRP Annals - Manufacturing Technology, 59, 2, pp. 740-759.

Martins, J., Lima, C., Martínez, H. and Grau, A. (2010). *A Matlab/Simulink framework for PLC controlled processes*, Matlab-Modelling, Programming and Simulations, p. 211.

Mathworks (2015). *Simulink PLC Coder - Generate IEC 61131-3 Structured Text for PLCs and PACs*, Available: http://uk.mathworks.com/products/sl-plc-coder/index.html?s_tid=gn_loc_drop [accessed February 17th, 2015]

Matley, J. (1969). *Keys to Successful Plant Startups*, Chemical Engineering, 76, 19, pp. 110-130.

Mayr, G. and Drath, R. (2007). *IEC PAS 62424 – Grafische Darstellung PLT-Aufgaben und Datenaustausch zu Engineering-Systemen / Representation of process control engineering requests in P&I Diagrams and data exchange to PCE-CAE tools*, atp - Automatisierungstechnische Praxis, 2007, 05, pp. 22-29.

McGregor, I. (2002). *The relationship between simulation and emulation*, Proceedings of the 2002 Winter Simulation Conference (WSC '02), San Diego, CA, USA, 2, pp. 1683-1688 vol.1682.

McGregor, I. (2012). *Introduction to Emulate3D: emulation, simulation, and demonstration*, Proceedings of the 2012 Winter Simulation Conference, (WSC '12), Berlin, Germany.

McGregor, I. (2015). *The Emulate3D Framework for the Emulation, Simulation and Demonstration of Industrial Systems*, Proceedings of the 2015 Winter Simulation Conference (WSC 2015), Huntington Beach, CA, USA, pp. 4101-4115.

McLean, C. and Leong, S. (2001). *The expanding role of simulation in future manufacturing*, Proceedings of the 33nd conference on Winter simulation (WSC '01), Arlington, Virginia, pp. 1478-1486.

Meier, H. and Kreusch, K. (2000). *Virtuelle Maschinen für eine realistische Simulation - Virtual machines for a realistic simulation*, wt Werkstattstechnik online, 90, 1/2, pp. 19-21.

Meinert, T. S., Don Taylor, G. and English, J. R. (1999). *A modular simulation approach for automated material handling systems*, Simulation Practice and Theory, 7, 1, pp. 15-30.

Mocanu, B., Tapu, R., Petrescu, T. and Tapu, E. (2011). *An experimental evaluation of 3D mesh decimation techniques*, 10th International Symposium on Signals, Circuits and Systems (ISSCS 2011), pp. 1-4.

Montalvo, B. and Phillips, R. (2010). *Using Emulation To Debug Control Logic Code: A Case Study*, Proceedings of the 2010 Winter Simulation Conference (WSC '10), Baltimore, Maryland, USA, pp. 1673-1677.

Montevechi, J. A. B., Pereira, T. F., Silva, C. E. S. D., Scheidegger, A. P. G. and Miranda, R. D. C. (2015). *Identification of the Main Research Methods Used in Simulation Projects*, Proceedings of the 2015 Winter Simulation Conference (WSC '15), Huntington Beach, USA, pp. 3469-3480.

Moore, P. R., Pu, J., Ng, H. C., Wong, C. B., Chong, S. K., Chen, X., Adolfsson, J., Olofsgard, P. and Lundgren, J. O. (2003). *Virtual engineering: an integrated approach to agile manufacturing machinery design and control*, Mechatronics, 13, 10, pp. 1105-1121.

Mueller, G. (2001a). *Emulation: using emulation to reduce commissioning costs on a high speed bottling line*, Proceedings of the 33nd Winter simulation conference (WSC '01), Arlington, Virginia, pp. 1461-1462.

Mueller, G. (2001b). *Optimizing PLC Controls on a High-Speed Bottling Line*, 2001 Symposium Proceedings, Brooks-PRI Automation, Inc. .

Mujber, T. S., Szecsi, T. and Hashmi, M. S. J. (2004). *Virtual reality applications in manufacturing process simulation*, Journal of Materials Processing Technology, 155-156, pp. 1834-1838.

Muller, D. (2012). *AutoMod: providing simulation solutions for over 30 years*, Proceedings of the 2012 Winter Simulation Conference (WSC '12), Berlin, Germany, p. 15.

Müür, M. and Pettai, E. (2010). *Analysis of Methods and Tools for Industrial Automation Engineering*, 8th International Symposium - "Topical Problems in the Field of Electrical and Power Engineering", Pärnu, Estonia, pp. 88-92.

Naake, J., Gellrich, A. and Kabitzsch, K. (2012). *Automatisierte Steuerungstests vereinfachen die virtuelle Inbetriebnahme in der Fabrikautomation*, atp edition – Automatisierungstechnische Praxis, 54, 4, pp. 14-16.

Neugebauer, R. and Schob, U. (2011). *Reducing the model generation effort for the virtual commissioning of control programs*, Production Engineering, 5, 5, pp. 539-547.

NVIDIA (2016). *PhysX SDK*, Available: https://developer.nvidia.com/physx-sdk [accessed July 13th, 2016]

Okino (2015). *CAD Conversions and CAD Data Re-purposing Through PolyTrans|CAD*, Available: http://www.okino.com/conv/conv.htm?0 [accessed February 10th, 2015]

OMG (2016). *Object Management Group - Business Process Model and Notation*, Available: http://www.bpmn.org/ [accessed June 19th, 2016]

OPC (2016a). *OPC Classic*, Available: https://opcfoundation.org/about/opc-technologies/opc-classic/ [accessed April 11th, 2016]

OPC (2016b). *OPC Unified Architecture*, Available: https://opcfoundation.org/about/opc-technologies/opc-ua/ [accessed April 11th, 2016]

Oppelt, M. and Urbas, L. (2014). *Integrated Virtual Commissioning an essential Activity in the Automation Engineering Process: From virtual commissioning to simulation supported engineering*, Industrial Electronics Society, IECON 2014 - 40th Annual Conference of the IEEE, pp. 2564-2570.

Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., Han, K. H. and Chang, M. (2006). *Development of Virtual Simulator for Visual Validation of PLC Program*, International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), p. 5.

Park, S. C. and Jang, J. S. (2011). *Virtual Plant for Control Program Verification*, Proceedings of International Conference on Circuits, System and Simulation (ICCSS 2011), pp. 290-294.

Park, S. C., Ko, M. and Chang, M. (2013). *A reverse engineering approach to generate a virtual plant model for PLC simulation*, The International Journal of Advanced Manufacturing Technology, 69, 9-12, pp. 2459-2469.

Pereira, A., Lima, C. and Martins, J. F. (2011). *The use of IEC 61131-3 to enhance PLC control and Matlab/Simulink process simulations*, Industrial Electronics (ISIE), 2011 IEEE International Symposium on, pp. 1243-1247.

Pfeifer, T. (2002). *Quality Management - Strategies, Methods, Techniques,* München, Germany, Carl Hanser Verlag GmbH & Co. KG, p.619.

PLCopen (2011). *Strong support for PLCopen XML specification*, Available: http://www.plcopen.org/pages/promotion/publications/downloads/press_releases/xml_support_nov2011.htm [accessed October 1st, 2013]

PLCopen (2013). *PLCopen adds independent schemes to IEC 61131-3*, Available: http://www.plcopen.org/pages/tc6_xml/xml_intro/index.htm [accessed October 1st, 2013]

Popovič, R., Trebuňa, P. and Kliment, M. (2015). *BASIC OVERVIEW ABOUT DIGITAL FACTORY AND VIRTUAL COMMISSIONING*, International Scientific Journal about Logistics, 2, 1, pp. 1-4.

Potthast, A. (2002). *Nutzungspotenziale von Virtual Reality für Werkzeugmaschinen*, Spectrum Research, UASA Hannover, pp. 44-47.

Preuße, S., Gerber, C. and Hanisch, H.-M. (2011). *Virtual Start-Up of Plants using Formal Methods*, International Journal of Computer Applications in Technology (IJCAT), 42, 2-3, pp. 108-126.

Pritschow, G. and Röck, S. (2004). *"Hardware in the Loop" Simulation of Machine Tools*, CIRP Annals - Manufacturing Technology, 53, 1, pp. 295-298.

ProSTEP iViP Association (2016). *STEP AP 242 Maintenance*, Available: http://www.prostep.org/en/projects/step-ap-242.html [accessed July 31st, 2016]

Pu, J. and Moore, P. R. (1998). *Towards Paradigm Shift in Machine-Design and Control*, 6th UK Mechatronics Forum International Conference (MECHATRONICS '98), Skövde, Sweden, pp. 23-30.

Pullan, T. T., Bhasi, M. and Madhu, G. (2010). *Application of concurrent engineering in manufacturing industry*, International Journal of Computer Integrated Manufacturing, 23, 5, pp. 425-440.

Putman, N. M., Maturana, F., Barton, K. and Tilbury, D. M. (2015). *Virtual Fusion: Integrating Virtual Components into a Physical Manufacturing System*, 15th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2015), 48, pp. 904-909.

Qiu, Z. M., Wong, Y. S., Fuh, J. Y. H., Chen, Y. P., Zhou, Z. D., Li, W. D. and Lu, Y. Q. (2004). *Geometric model simplification for distributed CAD*, Computer-Aided Design, 36, 9, pp. 809-819.

Quadros, W. R. and Owen, S. J. (2009). *Defeaturing CAD Models Using a Geometry-Based Size Field and Facet-Based Reduction Operators*, Proceedings of the 18th International Meshing Roundtable, pp. 301-318.

Quirós, G., Gora, M., Neidig, J. and Ermler, R. (2016). *Emulation von SPS auf virtueller Zeitbasis: Eine Voraussetzung für die virtuelle Inbetriebnahme*, Automation 2016, Baden-Baden, Germany, p. 12.

Rabe, M., Spieckermann, S. and Wenzel, S. (2008). *A new procedure model for verification and validation in production and logistics simulation*, Proceedings of the 2008 Winter Simulation Conference (WSC '08), Austin, TX, USA, pp. 1717-1726.

Raith, P. and Amman, W. (1992). *Erstellen und Testen von Ablaufvorschriften für Produktionssysteme*, ZWF-Zeitschrift für wirtschaftlichen Fabrikbetrieb, 87, 7, pp. 383-386.

Ramler, R., Putschögl, W. and Winkler, D. (2014). *Automated testing of industrial automation software: practical receipts and lessons learned*, Proceedings of the 1st International Workshop on Modern Software Engineering Methods for Industrial Automation, Hyderabad, India, pp. 7-16.

Reinhart, G. and Broy, M. (2003). *Teilprojekt MECHASOFT: Integrierte Entwicklung hochverfügbarer mechatronischer Systeme,* IN Broy, M., Eberspächer, J., Brügge, B., Färber, G., Reinhart, G. & Wildemann, H. (Eds.) Abschlussbericht FORSOFT II. Munich, Germany, Technische Universität München, pp.93-106.

Reinhart, G. and Wünsch, G. (2007). *Economic application of virtual commissioning to mechatronic production systems*, Production Engineering, 1, 4, pp. 371-379.

RIF - Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e. V. (2009). *Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e. V.*, Available: http://www.rif.fuedo.de [accessed 03. Jan. 2009]

RIF - Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e. V. (2012). *CIROS Engineering*, Available: http://www.ciros-engineering.com/en/home/ [accessed May 11th, 2012]

RIF e. V. - Institut für Forschung und Transfer (2013). *CIROS unlocks simulation and virtual reality for Autodesk Inventor*, Available: http://www.ciros-engineering.com/fileadmin/Templates/CIROS/Media/PRInventorPlugIn.pdf [accessed Dec. 15th, 2014]

RIF e.V. - Institut für Forschung und Transfer (2014). *CIROS - Virtual Reality*, Available: http://www.ciros-engineering.com/en/products/virtual-reality/ [accessed September 15th, 2014]

RIF e.V. - Institut für Forschung und Transfer (2015). *CIROS Planner*, Available: http://www.ciros-engineering.com/en/products/virtual-engineering/ciros-planner/ [accessed February 17th, 2015]

Robinson, S. (2006). *Conceptual Modeling for Simulation: Issues and Research Requirements*, Proceedings of the 2006 Winter Simulation Conference (WSC '06), pp. 792-800.

Romberg, A. and Haas, M. (2005). *Der Anlaufmanager : Effizient arbeiten mit Führungssystem und Workflow – Von der Produktidee bis zur Serie*, LOG_X, p.186.

Rossmann, J. and Heinze, F. (2010). *Modeling and simulation of malfunctions in automation systems*, Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on, pp. 1-8.

Rossmann, J., Schluse, M., Schlette, C. and Jung, T. (2006). *Virtual Human – der virtuelle Mensch in Fabriksimulationen,* Jahresbericht 2005/2006 des VRCA Virtual Reality Center Aachen, Infrastruktur, Projektaktivitäten, Veranstaltungen, Außendarstellung. Virtuelle Humanoide. Aachen, VRCA Virtual Reality Center Aachen, RWTH Aachen, pp.37-38.

Rossmann, J., Stern, O. and Wischnewski, R. (2007). *Eine Systematik mit einem darauf abgestimmten Softwarewerkzeug zur durchgängigen Virtuellen Inbetriebnahme von Fertigungsanlagen von der Planung über die Simulation zum Betrieb*, GMA-Kongress 2007, Baden-Baden, VDI-Berichte 1980, pp. 707-716.

Rossmann, J., Stern, O. and Wischnewski, R. (2012). *Virtuelle Inbetriebnahme von Transportsystemen*, automatisieren! by atp, pp. 13-26.

Rossmann, J., Wischnewski, R. and Stern, O. (2010). *A COMPREHENSIVE 3-D SIMULATION SYSTEM FOR THE VIRTUAL PRODUCTION*, The 8th annual Industrial Simulation Conference (ISC'2010), Budapest, Hungary, pp. 109-116.

Salamon, O. and Heidari, A. (2012). *Virtual commissioning of an existing manufacturing cell at Volvo Car Corporation using DELMIA V6,* Göteborg, Sweden, Chalmers University of Technology, Department of Signals and Systems - Automation Group, Master's Thesis, p.46.

Salt, J. D. (1993). *Simulation should be easy and fun!*, Proceedings of the 25th Winter simulation conference (WSC '93) Los Angeles, California, USA, pp. 1-5

Sargent, R. G. (2008). *Verification and validation of simulation models*, Proceedings of the 2008 Winter Simulation Conference (WSC '08), pp. 157-169.

Schetinin, N., Moriz, N., Kumar, B., Maier, A., Faltinski, S. and Niggemann, O. (2013). *Why do verification approaches in automation rarely use HIL-test?*, IEEE International Conference on Industrial Technology (ICIT 2013), Cape Town, South Africa, pp. 1428-1433.

Schleipen, M. and Drath, R. (2009). *Three-view-concept for modeling process or manufacturing plants with AutomationML*, Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on, pp. 1-4.

Schleipen, M., Drath, R. and Sauer, O. (2008). *The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system*, Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on, pp. 1786-1791.

Schleipen, M., Sauer, O. and Ammermann, C. (2010). *Die Anwendung der Digitalen Fabrik im laufenden Fabrikbetrieb*, Automation 2010, Baden-Baden, Germany, pp. 421-424.

Schlette, C. and Rossmann, J. (2009). *Robotics enable the simulation and animation of the Virtual Human*, Advanced Robotics, 2009. ICAR 2009. International Conference on, pp. 1-6.

Schlögl, W. (2007). *Einsatz der Digitalen Fabrik von der Anlagenplanung bis in den Laufenden Betrieb*, GMA-Kongress 2007, Baden-Baden, VDI-Berichte 1980, pp. 717-725.

Schlögl, W. (2012). *AutomationML and JT / STEP AP 242 XML - Survey and potential for cooperation*, Presentation, 2nd AutomationML User Conference, Sindelfingen, Germany, p. 21.

Schludermann, H., Kirchmair, T. and Vorderwinkler, M. (2000). *Soft-commissioning: Hardware-in-the-loop-based verification of controller software*, Proceedings of the 2000 Winter Simulation Conference, Vols 1 and 2, Orlando, Florida, USA, pp. 893-899.

Schulze, M. (2016). *Datennetz ist Schlüssel zur effizienten Produktion*, VDI Nachrichten, August 19th, 2016, VDI, Frankfurt/Main, Germany, pp. 12-13

Schumann, R. (2007). *Virtuelle Inbetriebnahme - Konzept und Realisierungsansatze / Virtual commissioning - Concept and realisation approaches*, GMA-Kongress 2007, Baden-Baden, VDI-Berichte 1980, pp. 697-706.

Seidel, S., Donath, U. and Haufe, J. (2012). *Towards an integrated simulation and virtual commissioning environment for controls of material handling systems*, Proceedings of the 2012 Winter Simulation Conference (WSC '12), Berlin, Germany, p. 12.

Shannon, R., E. (1976). *Simulation modeling and methodology*, Proceedings of the 76 Bicentennial conference on Winter simulation, Gaithersburg, MD, pp. 9-15.

Shannon, R. E. (1998). *Introduction to the Art and Science of Simulation*, Proceedings of the 1998 Winter Simulation Conference (WSC '98), 1, pp. 7-14 vol.11.

Shokry, H. and Hinchey, M. (2009). *Model-Based Verification of Embedded Software*, Computer, 42, 4, pp. 53-59.

Shuming, G. (2008). *Feature suppression based CAD mesh model simplification*, IEEE International Conference on Shape Modeling and Applications, Stony Brook, NY, USA pp. 271-272.

Siemens-Sinumerik (2014). *Hardware-in-the-loop shortens real commissioning*, SINUMERIK In-Sight 1/2014.

Siemens (2007). *Pictures of the Future: Zukunft der Fabriken – Fakten und Prognosen - Schneller, besser, billiger*, Available: http://www.siemens.com/innovation/de/publikationen/zeitschriften_pictures_of_the_future/pof_herbst_2007/zukunft_der_fabriken/fakten_und_prognosen.htm [accessed January 11th, 2015]

Siemens PLM (2014). *Shop Floor Program Validation - Advanced Machining Simulation Software*, Available: http://www.plm.automation.siemens.com/en_gb/products/nx/for-manufacturing/shop-floor-program-validation/index.shtml [accessed August 28, 2014]

Siemens (2016a). *Delivery Release: SIMULATION UNIT PB, SIMULATION UNIT PN128 and SIMULATION UNIT PN256*, Available: https://support.industry.siemens.com/cs/document/109476682/delivery-release%3A-simulation-unit-pb-simulation-unit-pn128-and-simulation-unit-pn256?dti=0&lc=en-DE [accessed May 5th, 2016]

Siemens (2016b). *SIMIT Simulation Framework V8.1 and SIMIT Virtual Controller V3.0*, Available: https://support.industry.siemens.com/cs/document/106448762/simit-simulation-framework-v8-1-and-simit-virtual-controller-v3-0-released-for-ordering-and-delivery?dti=0&lc=en-WW [accessed May 5th, 2016]

SimPlan (2016). *Simulation tools - It has to be the right software*, Available: https://www.simplan.de/en/software/tools.html [accessed Feb. 9th, 2016]

Smith, J. S. and Cho, Y. (2008). *OFFLINE COMMISSIONING OF A PLC-BASED CONTROL SYSTEM USING ARENA*, Proceedings of the 2008 Winter Simulation Conference (WSC '08), pp. 1802-1810.

Sohlenius, G. (1992). *Concurrent Engineering*, CIRP Annals - Manufacturing Technology, 41, 2, pp. 645-655.

SolidWorks (2013). *SolidWorks Labs - Collada Export*, Available: http://labs.solidworks.com/products/product.aspx?name=colladaexport [accessed Feb. 28th, 2013]

Spath, D. and Landwehr, R. (2000). *Three-Dimensional Programming and Simulation of PLC-Controlled Manufacturing Systems*, Journal for Manufacturing Science and Production, 3, pp. 189-194.

Spath, D. and Osmers, U. (1996). *Virtual reality - An approach to improve the generation of fault free software for programmable logic controllers (PLC)*, Engineering of Complex Computer Systems, 1996. Proceedings., Second IEEE International Conference on, pp. 43-46.

Starner, C. and Chessin, M. (2010). *Using Emulation to Enhance Simulation*, Proceedings of the 2010 Winter Simulation Conference (WSC '10), Baltimore, Maryland, USA, pp. 1711-1715.

Staudinger (2015a). *Individualmodell - Sintek-Avex Corporation -Taiwan 2001*, Available: http://www.staudinger-est.de/simulation/referenzen/documents/Taiwandt.pdf#taiwan [accessed Dec. 1st, 2015]

Staudinger (2015b). *Simulation - Models made of Fischertechnik*, Available: http://www.staudinger-est.de/en/simulation/ [accessed Dec. 1st, 2015]

Stern, O., Hoffmann, P. and Schumann, R. (2010). *KMU und VIBN in der Fertigungstechnik*, Presentation, Meeting of VDI/VDE-GMA Committee FA 6.11 "CACE", Dec. 1st, 2010, VDI/VDE Gesellschaft Mess- und Automatisierungstechnik (GMA), ABB Ladenburg, Germany

Stich, P. and Reinhart, G. (2013). *Mechatronic sketching of manufacturing systems using Physically Based Models A novel approach for simulation-based systems engineering*, IEEE Symposium on Industrial Electronics and Applications (ISIEA 2013), Kuching, Malaysia, pp. 1-6.

Strahilov, A., Mrkonjic, M. and Kiefer, J. (2012). *Development of 3D CAD simulation models for virtual commissioning*, Proceedings of the Ninth International Symposium on Tools and Methods of Competitive Engineering (TMCE 2012), Karlsruhe, Germany, pp. 1281-1287.

Sturrock, D. T. (2012). *Tutorial: Tips for successful practice of simulation*, Proceedings of the 2012 Winter Simulation Conference (WSC '12), Berlin, pp. 1-8.

Suh, S.-H., Seo, Y., Lee, S.-M., Choi, T.-H., Jeong, G.-S. and Kim, D.-Y. (2003). *Modelling and Implementation of Internet-Based Virtual Machine Tools*, The International Journal of Advanced Manufacturing Technology, 21, 7, pp. 516-522.

Sutherland, I. E. (1965). *The Ultimate Display*, Proceedings of IFIP Congress, New York, USA, pp. 506–508.

Sutherland, I. E. (1968). *A head-mounted three dimensional display*, Proceedings of the fall joint computer conference, part I, San Francisco, California, pp. 757-764.

Syska, B. (2004). *Structured approach to industrial control system design,* Pontypridd, Wales, UK, University of Glamorgan, School of Technology, PhD Thesis, p.154.

Taksale, A., Vaidya, V., Shahane, P., Dronamraju, G. and Deulkar, V. (2015). *Low cost hardware-in-loop for automotive application*, Industrial Instrumentation and Control (ICIC), 2015 International Conference on, pp. 1109-1114.

Thakur, A., Banerjee, A. G. and Gupta, S. K. (2009). *A survey of CAD model simplification techniques for physics-based simulation applications*, Computer-Aided Design, 41, 2, pp. 65-80.

Thapa, D., Park, C. M., Dangol, S. and Wang, G.-N. (2006). *III-Phase Verification and Validation of IEC Standard Programmable Logic Controller*, International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), p. 5.

Thieme, J. and Hanisch, H. M. (2002). *Model-based generation of modular PLC code using IEC61131 function blocks*, Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on, 1, pp. 199-204 vol.191.

Thramboulidis, K. (2008). *Challenges in the development of Mechatronic systems: The Mechatronic Component*, Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on, pp. 624-631.

Thron, M., Zipper, H., Magnus, S., Süß, S., Göbeler, C., Liu, Z. and Diedrich, C. (2016). *Beschreibung des normalen und gestörten Verhaltens mechatronischer Komponenten für den automatisierten virtuellen Anlagentest*, Automation 2016, Baden-Baden, Germany, p. 12.

Traceparts (2014). *Download free CAD drawings, technical data & 3D models*, Available: http://www.traceparts.com/use-and-manage-3d-cad-models/download-free-cad-models/ [accessed September 29th, 2014]

Trimble (2014). *3D Warehouse*, Available: https://3dwarehouse.sketchup.com/index.html [accessed September 29th, 2014]

TÜV Rheinland (2014). *Factory Acceptance Test*, Available: http://www.tuv.com/en/corporate/business_customers/materials_testing_and_inspection/supply_chain_services/factory_acceptance_test/factory_acceptance_test.html [accessed August 22, 2014]

Tecnomatix (2005). *eM-PLC and STEP 7 Professional – eMPower for manufacturing process management / Model-based, automatic PLC program generation, simulation and verification*, fact sheet, p. 3.

University of Cambridge (2014). *About Shrinkwrap Models*, Available: http://www3.eng.cam.ac.uk/DesignOffice/cad/proewild3/usascii/proe/asm/asm_three_sub/about_shrinkwrap_models.htm [accessed September 20th, 2014]

Van der Wal, E. (1999). *Introduction into IEC 1131-3 and PLCopen*, The Application of IEC 61131 to Industrial Control: Improve Your Bottom Line Through High Value Industrial Control Systems (Ref. No. 1999/076), IEE Colloquium on, pp. 2/1-2/8.

VDI 4499, (2008). *VDI-Richtlinie 4499 - Blatt 1 / Part 1: Digitale Fabrik - Grundlagen / Digital factory - Fundamentals*, p. 52.

VDI 3633, (2010). *VDI-Richtlinie 3633 - Entwurf / Draft, Blatt 1 / Part 1:Simulation von Logistik-, Materialfluss und Produktionssystemen - Grundlagen / Simulation of systems in materials handling, logistics and production - Fundamentals*, p. 49.

VDI 4499, (2011). *VDI-Richtlinie 4499 - Blatt 2 / Part 2: Digitale Fabrik - Digitaler Fabrikbetrieb / Digital Factory - Digital Factory Operations*, p. 51.

VDI 3693, Barth, M., Hefner, F., Hoernicke, M., Hoffmann, P., Hundt, L., Iffländer, B., Jost, H., Krause, H., Liu, Z., Oppelt, M., Puntel-Schmidt, P., Schumann, R., Stern, O., Strigl, T., Wolf, G. (2015a). *VDI-Richtlinie 3693, Blatt 1 - Virtuelle Inbetriebnahme - Modellarten und Glossar / VDI Guideline 3693, Part 1: Virtual Commissioning - Model types and glossary*, p. 23.

VDI 4499, (2015b). *VDI-Richtlinie 4499 - Blatt 4 / Part 4: Digitale Fabrik - Ergonomische Abbildung des Menschen in der Digitalen Fabrik / Digital factory - Ergonomic representation of humans in the digital factory*, p. 54.

VDI/VDE 3695, (2010). *VDI/VDE-Richtlinie 3695 - Blatt 1 / Part 1: Engineering von Anlagen / Engineering of industrial plants*, p. 12.

VDW (1997). *Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen, VDW-Bericht 0162, WZL - Werkzeugmaschinenlabor der RWTH Aachen*.

Vepsäläinen, T. and Kuikka, S. (2014). *Integrating model-in-the-loop simulations to model-driven development in industrial control*, simulation, 90, 12, pp. 1295-1311.

Verl, A., Fritsch, D. (2008). *Steuerungsentwicklung mit Simulationssoftware*, wt Werkstattstechnik online, 98, 5, pp. 370-376.

Viswanathan, J., Harrison, W., Tilbury, D. and Gu, F. (2011). *Using Hybrid Process Simulation to Evaluate Manufacturing System Component Choices: Integrating a Virtual Robot with Physical System*, Proceedings of the 2011 Winter Simulation Conference (WSC '11), Phoenix, AZ, USA, pp. 2827-2838.

Vorderwinkler, M., Eder, T., Steringer, R. and Schleicher, M. (1999). *An architecture for Soft-Commissioning - Verifying control software by linking discrete event simulators to real world control systems*, 13th European Simulation Multiconference (ESM'99), Warsaw, Poland, pp. 191-198.

Warnecke, H.-J. and Schraft, R. D. (1984-1997). *Handhabungstechnik,* Handbuch Handhabungs-, Montage- und Industrierobotertechnik. Landsberg am Lech, Germany, Verl. Moderne Industrie.

Weber, K. H. (2006). *Inbetriebnahme verfahrenstechnischer Anlagen : Praxishandbuch mit Checklisten und Beispielen,* Berlin, Heidelberg, Springer, p.397.

Weidemann, D. (2008). *The power of visions - complete plant descriptions in a neutral data format*, Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on, Dresden, Germany, p. 201.

Wenk, M. (2008). *Virtuelle Inbetriebnahme von Produktionsanlagen - Aufwand-Nutzen, Umsetzungsstrategien, zukünftige Entwicklungen*, Tagungsband SPS/IPC/Drives 2008, Nürnberg, pp. 531-539.

Wenzel, S., Boyaci, P. and Jessen, U. (2010). *Simulation in Production and Logistics: Trends, Solutions and Applications,* IN Dangelmaier, W., Blecken, A., Delius, R. & Klöpfer, S. (Eds.) Advanced Manufacturing and Sustainable Logistics*.* Springer Berlin Heidelberg, pp.73-84.

Westkämper, E., Bierschenk, S. and Kuhlmann, T. (2003). *Digitale Fabrik – nur was für die Großen? / Digital Manufacturing – only for large scale enterprises?*, wt Werkstattstechnik online, 93, 1/2, pp. 22-26.

Westkämper, E., Niemann, J., Warschat, J., Scheer, A.-W., Thomas, O., Bullinger, H.-J., Spath, D. and Warnecke, H.-J. (2009). *Methoden der digitalen Planung,* IN Bullinger, H.-J. (Ed.) Handbuch Unternehmensorganisation*.* Springer Berlin Heidelberg, pp.515-568.

Weyrich, M., Klein, P. and Steden, F. (2014). *Reuse of modules for mechatronic modeling and evaluation of manufacturing systems in the conceptual design and basic engineering phase*, 19th IFAC World Congress, Cape Town, South Africa, pp. 3450-3455.

Wiendahl, H.-P., Harms, T. and Fiebig, C. (2003). *Virtual factory design—a new tool for a co-operative planning approach*, Int. Journal of Computer Integrated Manufacturing, 16, 7/8, pp. 535–540.

Wiendahl, H.-P., Hegenscheidt, M. and Winkler, H. (2002). *Anlaufrobuste Produktionssysteme - Ramp-up-sturdy production systems*, wt Werkstattstechnik online, 92, 11/12, pp. 650-655.

Wikander, J. and Törngren, M. (1998). *Mechatronics as an Engineering Science*, Proc. of the 6th UK Mechatronics Forum Int. Conf., Skövde Sweden, p. 6.

Wikipedia (2014). *Cave Automatic Virtual Environment*, Available: http://de.wikipedia.org/wiki/Cave_Automatic_Virtual_Environment [accessed Sept. 11, 2014]

Wikipedia (2016). *List of discrete event simulation software*, Available: https://en.wikipedia.org/wiki/Discrete_event_simulation [accessed Feb. 9th, 2016]

Williams, E. and Ülgen, O. M. (2001). *Statistics and Operations Research and Optimization - Simulation Methodology, Tools, and Applications* IN Zandin, K. B. (Ed.) Maynard's Industrial Engineering Handbook*.* 5th ed., McGraw-Hill Education, pp.(11)101-119.

Wilson, J. M. (2003). *Gantt charts: A centenary appreciation*, European Journal of Operational Research, 149, 2, pp. 430-437.

WinMOD (2016). *WinMOD Configurations Data Sheets*, Available: http://www.winmod.de/en/index.php?page=winmod-konfigurationen [accessed May 1st, 2016]

Wischnewski, R. (2007). *Virtuelle Inbetriebnahme*, A&D Kompendium 2007/2008, pp. 64-66.

Wischnewski, R. and Freund, E. (2004). *COSIMIR Transport: Modeling, simulation and emulation of modular carrier based transport systems*, International Conference on Robotics & Automation, New Orleans, LA, United States, 2004, pp. 5171-5176.

Wischnewski, R. and Rossmann, J. (2010). *Modelling of Track Bound Transport Systems for the Virtual Production*, 21st IASTED International Conference Modelling and Simulation (MS 2010), Banff, pp. 214-221.

Yong, T. and Hongbin, G. (2010). *CAD Model's Simplification and Conversion for Virtual Reality*, Information and Computing (ICIC), 2010 Third International Conference on, 4, pp. 265-268.

Zäh, M. F., Lercher, B., Pörnbacher, C. and Wünsch, G. (2003). *Datenmanagement in der Mechatronik*, wt Werkstattstechnik online, 93, 7/8, pp. 541-544.

Zäh, M. F. and Pörnbacher, C. (2008). *Model-driven development of PLC software for machine tools*, Prod. Eng. Res. Devel., 2, pp. 39-46.

Zäh, M. F., Pörnbacher, C. and Milberg, J. (2005). *A model-based method to develop PLC software for machine tools*, CIRP Annals - Manufacturing Technology, 54, 1, pp. 371-374.

Zäh, M. F. and Wünsch, G. (2005). *Schnelle Inbetriebnahme von Produktionssystemen*, wt Werkstattstechnik online, 95, 9, pp. 699-704.

Zäh, M. F., Wünsch, G., Hensel, T. and Lindworsky, A. (2006a). *Feldstudie – Virtuelle Inbetriebnahme / Experimental study on virtual commissioning*, wt Werkstattstechnik online, 96, 10, pp. 767-771.

Zäh, M. F., Wünsch, G., Hensel, T. and Lindworsky, A. (2006b). *Nutzen der virtuellen Inbetriebnahme: Ein experiment - Use of virtual commissioning: An experiment*, ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb, 101, 10, pp. 595-599.

Zeugträger, K. (1998). *Anlaufmanagement für Großanlagen,* Hannover, Germany, Universität Hannover, Institut für Fabrikanlagen (IFA), Dissertation, p.157.

Zhou, M. and Wang, M. Y. (2012). *Engineered Model Simplification for Simulation Based Structural Design*, Computer-Aided Design & Applications, 9, 1, pp. 87-94.

Publications

Papers, presentations (not appended) and VDI Guideline (not appended)

- Hoyer, M., Schumann, R., Hoffmann, P. and Premier, G. C. (2008). Virtuelle Inbetriebnahme mit ModelCAT - Vom Prototypen zum industriellen Einsatz / Virtual Start-Up with ModelCAT - From Prototypical Realisation to Industrial Implementation, Automation 2008, 3rd and 4th June 2008, Baden-Baden, VDI-Berichte 2032, pp. 203-206.

- Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2009). Analyse und Konzepte zur vereinfachten Virtuellen Inbetriebnahme von Fertigungssystemen - Analysis and concepts for simplified virtual commissioning of manufacturing systems, Automation 2009, Baden-Baden, VDI-Berichte 2067, pp. 481-484.

- Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2010). Virtual Commissioning of Manufacturing Systems - A Review and new Approaches for Simplification, Proceedings of the 24th European Conference on Modelling and Simulation, June 1st - 4th, 2010, Kuala Lumpur, Malaysia, pp. 175-181.

- Stern, O., Hoffmann, P. and Schumann, R. (2010). KMU und VIBN in der Fertigungstechnik, Presentation, Meeting of VDI/VDE-GMA Committee FA 6.11 "CACE", VDI/VDE Gesellschaft Mess- und Automatisierungstechnik (GMA), Dec. 1st, 2010, Ladenburg, Germany

- Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C. (2012). Research on simplified modelling strategy for virtual commissioning, The 24th European Modeling & Simulation Symposium (Simulation in Industry) EMSS 2012, September, 19-21, 2012, Vienna, Austria, pp. 293-302

- Hoffmann, P. (2013). Modellierung für die Virtuelle Inbetriebnahme (VIBN) - Verhaltensmodellierung Transportwagen, Presentation, AutomationML-Workshop "Use-Cases for FBD", June 19th, 2013, Phoenix Contact, Bad Pyrmont, Germany

- Barth, M., Hefner, F., Hoernicke, M., Hoffmann, P., Hundt, L., Iffländer, B., Jost, H., Krause, H., Liu, Z., Oppelt, M., Puntel-Schmidt, P., Schumann, R., Stern, O., Strigl, T., Wolf, G. (2015). VDI-Richtlinie 3693, Blatt 1 - Virtuelle Inbetriebnahme - Modellarten und Glossar / VDI Guideline 3693, Part 1: Virtual Commissioning - Model types and glossary, p. 23.

Virtuelle Inbetriebnahme mit Model^{CAT}

Vom Prototypen zum industriellen Einsatz

Virtual Start-Up with Model^{CAT}

From Prototypical Realisation to Industrial Implementation

Dr. **M. Hoyer**, Prof. Dr.-Ing. **R. Schumann**,
Dipl.-Ing. **P. Hoffmann**, Fachhochschule Hannover;
Dr. **G.C. Premier**, University of Glamorgan, Pontypridd (UK)

Kurzfassung

Model^{CAT} ist ein Konzept zur automatischen Modellgenerierung, Simulation und Funktionsprüfung für verfahrenstechnische Anlagen in der Endphase der Anlagenplanung. In diesem Beitrag werden Erkenntnisse der prototypischen Umsetzung von Model^{CAT} dazu genutzt, um Empfehlungen für eine industrielle Realisierung des Konzeptes zu entwickeln. Damit rückt das Ziel, mit Hilfe einer virtuellen Inbetriebnahme Kosten und Zeiten der realen Inbetriebnahme zu reduzieren sowie die Qualität der Anlage zu verbessern, in greifbare Nähe.

Abstract

Model^{CAT} is a concept for the automatic model generation, simulation and function test for chemical processes at the end of detailed engineering. In this paper the experiences gained from the prototypical realisation of Model^{CAT} are used to extrapolate to the requirements for an industrial implementation. The main goals of the Model^{CAT} approach to minimise time and money for the real start-up and to increase the plant's quality could thus be realised in the near future.

1 Einführung

Für die virtuelle Inbetriebnahme einer Anlage werden simulierbare (und aussagefähige) Modelle des verfahrenstechnischen Prozesses sowie des Prozessleitsystems benötigt, die in einer leistungsfähigen Simulationsumgebung gemeinsam simuliert werden können, siehe z.B. [1]. Während es bei den Prozessleitsystemen bereits Systememulatoren gibt, auf denen die Original-Leitsystemprogramme ablaufen können, ist die Entwicklung von aussagefähigen Modellen für den verfahrenstechnischen Prozess nach wie vor ein aufwändiger und kostenintensiver Prozess, der Expertenwissen im Bereich verfahrenstechnischer Simulationsmodelle sowie der entsprechenden Simulationsumgebungen erfordert.

Für die Generierung von Simulationsmodellen für eine verfahrenstechnische Anlage setzt Model^{CAT} bei den vollständigen Planungsinformationen an, die in einem CAE-Planungssystem nach Abschluss der Anlagenplanung zusammengeführt sind. Auf der Basis dieser Planungsinformationen wird die Anlage später tatsächlich gebaut, daher sollte – zumindest theoretisch – diese Information auch für die Generierung eines simulierbaren Anlagenmodells – der „Virtuellen Anlage“ – ausreichen.

In diesem Beitrag werden die Arbeitsschritte beschrieben, die notwendig werden, um das Model^{CAT}-Konzept für eine industrielle Realisierung weiterzuentwickeln, basierend auf den gewonnenen Erfahrungen und Erkenntnissen bei der prototypischen Umsetzung des Model^{CAT}-Konzeptes [2, 3]. Dabei wird zunächst das Model^{CAT}-Konzept skizziert. Anschließend erfolgt die Beschreibung des aktuellen Demonstrations-Prototyps von Model^{CAT}. Die bei der prototypischen Umsetzung gewonnenen Erfahrungen dienen im Folgenden als Ausgangspunkt für die Diskussion der Aufgabestellungen bei einer industrielle Realisierung von Model^{CAT}. In der Abschlussbetrachtung werden die Perspektiven einer industriellen Realisierung des Model^{CAT}- Konzeptes zusammengefasst.

2 Das Model^{CAT}-Konzept

Model^{CAT} beschreibt einen systematischen Ansatz zur automatischen Generierung von Anlagensimulationsmodellen basierend auf den Ergebnissen der CAE-Anlagenplanung und mit

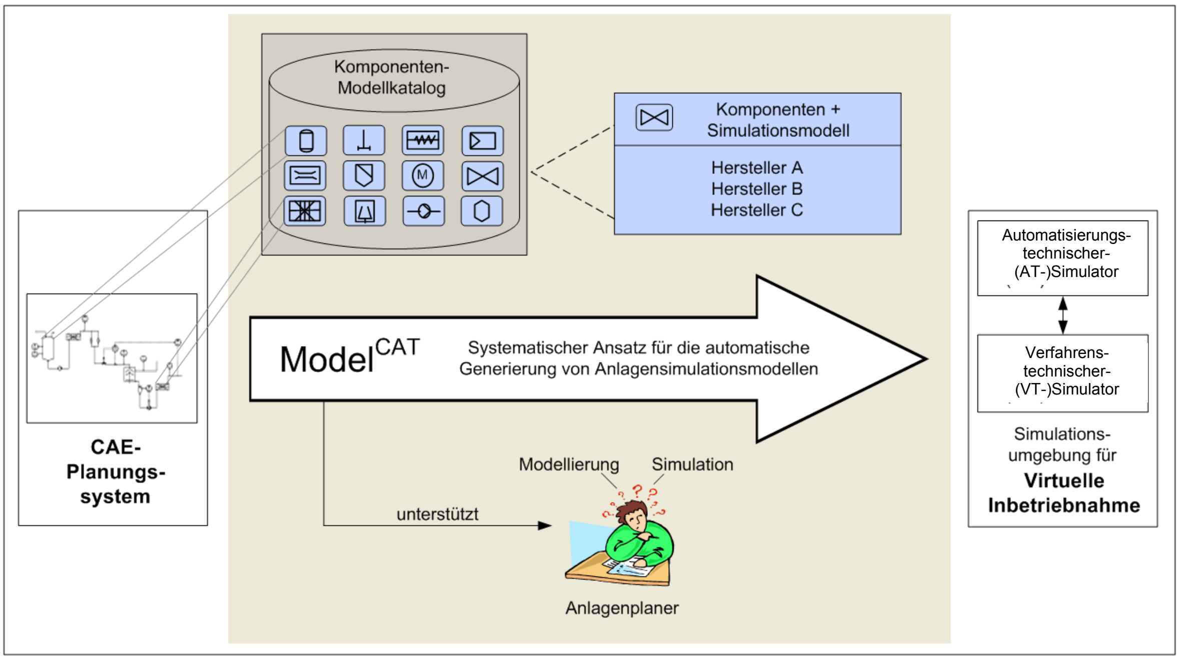

Komponenten-Modellkatalog
Komponenten + Simulationsmodell
Hersteller A
Hersteller B
Hersteller C
Automatisierungs-technischer-(AT-)Simulator
Verfahrens-technischer-(VT-)Simulator
Simulations-umgebung für
Virtuelle Inbetriebnahme
ModelCAT
Systematischer Ansatz für die automatische Generierung von Anlagensimulationsmodellen
CAE-Planungs-system
Modellierung
Simulation
unterstützt
Anlagenplaner

Bild 1: Konzept der automatischen Modellgenerierung mit Model^{CAT}

Hilfe von Simulationsmodellen der (physikalischen) Komponenten [2, 3]. Die Eckpfeiler des Model^{CAT}-Konzeptes sind (Bild 1): das CAE-Planungsystem, der Komponentenmodellkatalog, die Simulationsumgebung mit dem AT-Simulator für das Automatisierungssystem und dem VT-Simulator für den verfahrenstechnischen Prozess, sowie das Modell-aggregationsmodul MAM, das automatisch die Anlagensimulationsmodelle für die jeweiligen Simulatoren auf Basis der Anlagenplanungsdaten und des Komponenten-Modellkatalogs generiert. Die Unterstützung des Anlagenplaners bei dieser Aufgabe erfolgt über eine graphische Benutzeroberfläche (GUI = Graphical User Interface).

3 Model^{CAT}-Demonstrations-Prototyp

Mit dem auf der Basis des Model^{CAT}-Konzeptes entwickelte Funktions-Prototypen konnte demonstriert werden, wie eine Anlagensimulation für die virtuelle Inbetriebnahme aus dem R&I-Fließbild des CAE-Planungssystems COMOS [4] heraus generiert werden kann, Bild 2. Dazu wird eine vereinfachte Simulationsumgebung genutzt, die aus gPROMS [5] als verfahrenstechnischem Simulator und aus Matlab-Simulink [6] als automatisierungstechnischem Simulator besteht. Grundlage der automatisierten Modellgenerierung sind die Anlagenplanungsdaten, die nach Abschluss des Detail Engineering in der COMOS-Datenbank vorliegen.

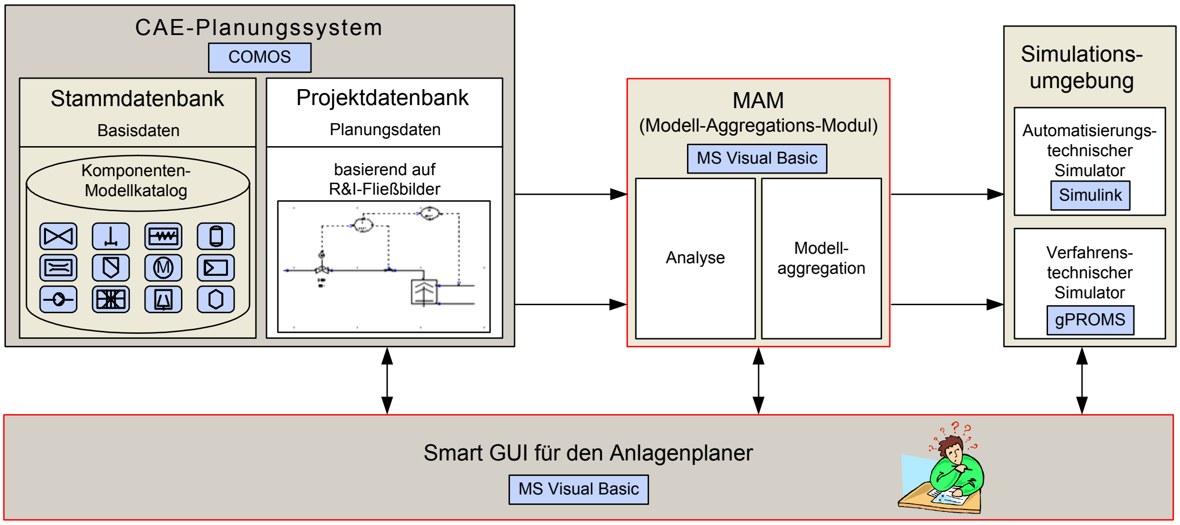

CAE-Planungssystem
COMOS
Stammdatenbank
Basisdaten
Komponenten-Modellkatalog
Projektdatenbank
Planungsdaten
basierend auf R&I-Fließbilder
MAM
(Modell-Aggregations-Modul)
MS Visual Basic
Analyse
Modell-aggregation
Simulations-umgebung
Automatisierungs-technischer Simulator
Simulink
Verfahrens-technischer Simulator
gPROMS
Smart GUI für den Anlagenplaner
MS Visual Basic

Bild 2: Aufbau des Model^{CAT}-Prototyps

In Bild 3 ist der prinzipielle Ablauf einer Anlagensimulation mit Model^{CAT} dargestellt:

(1) Zunächst wählt der Anlagenplaner im R&I-Fließbild von COMOS die zu simulierende Teilanlage aus.

(2) Model^{CAT} analysiert den gewählten Ausschnitt in Bezug auf die Grenzen und die Aufteilung in einen automatisierungstechnischen und einen verfahrenstechnischen Bereich, im

Spezifikationsfenster werden dazu die notwendigen Randbedingungen für die Simulation ermittelt.

(3) Die Simulationsmodelle werden dann mit Hilfe des Modell-Aggregationsmoduls (MAM) aus den Komponentenlisten und Verbindungslisten automatisch zusammengestellt, die wichtigste Voraussetzung dafür ist der Zugriff auf alle benötigten Simulationsmodelle der Komponenten, die in der COMOS-Datenbank hinterlegt sind. Dabei werden zum einen das verfahrenstechnische Prozessmodell für den zu simulierende Teilprozess im gPROMS-Format sowie ein Modell der Automatisierungsfunktionen im Simulink-Format automatisch generiert. Die Schnittstelle zwischen beiden Modellen bilden die Sensoren und Aktoren, die in beiden Modellen erscheinen: als Informationsquellen und – senken im Simulink-Modell, als physikalische Messfühler und Stellglieder im gPROMS-Modell.

(4) Die Simulation wird schließlich vom Anlagenplaner manuell gestartet, der dann über Simulink auch eine Darstellung der Simulationsergebnisse erstellen kann.

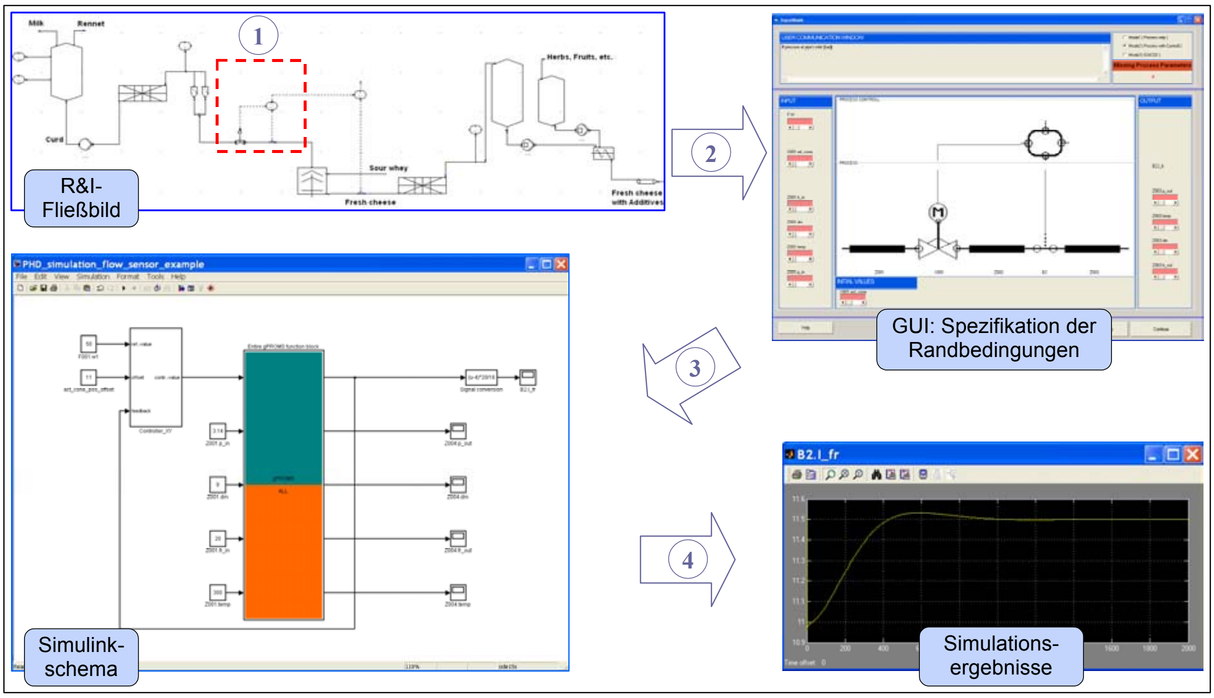


Bild 3: Prinzipieller Ablauf einer Anlagensimulation mit Model^{CAT}

Diese Vorgehensweise wurde mit dem Model^{CAT}-Prototyp am Beispieles einer Frischkäse-Produktionsanlage exemplarisch durchgeführt [7]. Dazu wurden verschiedene Schaltungsvarianten im R&I-Fliessbild aufgebaut und mit Hilfe von Model^{CAT} simuliert. So konnten die zu erwartenden Funktionseigenschaften analysiert und die optimale Verschaltung bestimmt werden.

4 Industrielle Realisierung von Model^{CAT}

Die bei der prototypischen Umsetzung gewonnenen Erfahrungen dienen im Folgenden als Ausgangspunkte für Vorschläge zur Weiterentwicklung von Model^{CAT} für eine industrielle Realisierung. Dabei werden speziell die folgenden Aufgabenstellungen betrachtet:

- Aufbau eines umfassenden Komponenten-Modellkataloges
- Erweiterungen der Simulationsumgebung
- Einführung von Schnittstellen-Standards
- Erweiterung des GUI
- Rolle des Anlagenplaners / Einbindung des Inbetriebnehmers und interaktiver Inbetriebnahmeszenarien

4.1 Aufbau eines umfassenden Komponenten-Modellkataloges

Die Voraussetzung für eine automatische Generierung von Anlagensimulationsmodellen am Ende des Detail Engineering ist die Verfügbarkeit von Simulationsmodellen der verwendeten Anlagenkomponenten. Die wenigen Komponentenmodelle, die für die Demonstration des Model^{CAT}-Konzeptes erstellt wurden, müssen bei einer industriellen Realisierung durch einen systematisch aufgebauten umfassenden Modellkatalog der bei der Planung von industriellen Anlagen zum Einsatz kommenden marktgängigen Komponenten ersetzt werden.

Die Komponenten-Simulationsmodelle sollten dabei vornehmlich von den Komponentenherstellern zusammen mit den technischen Daten der Komponenten geliefert werden. Um die Komponentenhersteller dazu zu bewegen, wird allerdings ein motivierender Druck von Seiten ihrer Kunden, insbesondere den großen Anlagenbetreibern und -planern, sicher notwendig werden.

Alternativ kann ein neutraler Modellentwickler die Rolle des Modelllieferanten übernehmen, z.B. in Form von Modell-Entwicklungsfirmen in Kooperation mit Hochschulen. Vorteile dieser Alternative wäre die neutrale und unabhängige Sicht auf das Verhalten der Komponenten. Auch dieses Vorgehen erfordert jedoch die Kooperation der Komponentenhersteller.

- **Modellqualität**

Während in der prototypischen Umsetzung ausschließlich White-Box-Modelle entwickelt und verwendet wurden, muss eine industrielle Umsetzung auch Black-Box-Modelle berücksichtigen/integrieren können, da nicht jeder Hersteller bereit sein wird, die internen Details seiner Komponenten preiszugeben. Die Qualität der Simulationsmodelle spielt eine entscheidende Rolle in Bezug auf die Aussagekraft der Simulationsergebnisse. Daher ist dafür zu sorgen, dass Informationen über die Modellgenauigkeit und zum Gültigkeitsbereich gemeinsam mit den Modellen geliefert und im Modellkatalog abgelegt werden. Bei der Kombination der Modelle zu komplexen Simulationsschaltungen ist entscheidend, ob sich mit den angewandten Aggregationsmethoden und Simulationsalgorithmen noch ausreichend robuste Simulationen mit zuverlässigen Ergebnissen durchführen lassen. Diesen Aspekten kommt in der industriellen Realisierung insbesondere deshalb eine große Bedeutung zu, weil jetzt Modellentwickler und –nutzer nicht mehr identisch sind. Die Modellentwickler sind verantwortlich für Genauigkeit und Gültigkeitsbereich der Modelle, die implementierten Aggregationsmethoden und Simulationsalgorithmen müssen auch ohne Nutzereingriff für ein Höchstmass an Robustheit der Simulation und Zuverlässigkeit der Simulationsergebnisse auch bei komplexen Anlagenmodellen sorgen.

Von der Qualität der Modelle wird es abhängen, welche Aussagen aus den Anlagensimulationen bei einer virtuellen Inbetriebnahme gewonnen werden können. Dabei ist absehbar, dass für die Entdeckung der meisten Auslegungsfehler zunächst relativ ungenaue Komponentenmodelle ausreichen werden. Bei der Erstellung von Modellkatalogen sollte daher im ersten Schritt die Modellgenauigkeit nicht ganz im Vordergrund stehen, sondern zunächst die Bereitstellung eines möglichst umfassenden Modellkatalogs, in dem Simulationsmodelle aller marktgängigen und für die Anlagensimulation benötigten Komponenten bereitgestellt werden. Die Detaillierung der Simulationsmodelle kann dann über die Jahre weiter entwickelt werden, so wie dies bereits im Bereich der Simulation von elektronischen Schaltungen für die Platinenherstellung in den letzten Jahrzehnten mit Erfolg vorgeführt wurde, siehe [8].

- **Integration des Modellkataloges in CAE-Planungssystem**

Im Prinzip sind zwei Möglichkeiten der Integration von Modellen in das CAE-Planungssystem vorstellbar: Zum einen können die Komponenten-Modelle in der Stammdatenbank des CAE-Planungssystems datentechnisch an die jeweilige Anlagen-Komponente angehängt werden. Diese Methode wurde für den Model^{CAT}-Prototyp verwendet, was jedoch Nachteile in Bezug auf den fortlaufenden Aufwand für die Pflege der Datenbank des CAE-Planungssystems hat. Aus diesem Grund scheint die zweite Alternative viel versprechender: Ein separater eigenständiger Modellkatalog, zu dem ein Link (Verweis) von den Anlagen-Komponenten in der COMOS-Datenbank hergestellt wird. Ein solcher eigenständiger Modellkatalog kann z.B. mit einem Datawarehouse für Komponentenmodelle realisiert werden, auch eine Internetplattform ist denkbar, siehe z.B. [9]. Voraussetzung für einen effektiven Austausch von Komponentenmodellen ist allerdings, dass dazu standardisierte Informationsschnittstellen genutzt werden können, die eine manuelle Einbringung und Pflege der Modelle in Modelldatenbanken vermeidet und einen standardisierten Zugang zu den Modellen ermöglicht.

4.2 Erweiterungen der Simulationsumgebung

Obwohl die Wahl der Simulationstools für die prototypische Realisierung aufgrund von Budgetlimitierungen beschränkt war, konnte die generelle Machbarkeit des Model^{CAT}-Konzeptes demonstriert werden. Für einen industriellen Ansatz sollte der AT-Simulator Matlab/Simulink durch ein emuliertes PLS ersetzt werden, womit eine „1zu1“ Umsetzung der geplanten PLS-Funktionen direkt und ohne Neu-Modellierung gewährleistet werden kann. In der prototypischen Realisierung wurde der VT-Simulator gPROMS als Modellblock in Simulink eingebettet/verwendet. Die gO:Simulink-Schnittstelle [10] ermöglicht dabei allerdings keinen einfachen Zugriff auf die innerhalb von gPROMS simulierten Funktionen über die aus dem gPROMS-Block explizit herausgeführten Signale hinaus. Der VT-Simulator in einer industriellen Realisierung sollte daher zum einen auf einer objektorientierten Modellstruktur aufbauen und zum anderen Möglichkeiten zur graphischen Darstellung des Simulationsmodells und der Simulationsergebnisse bieten, die eine detaillierte Untersuchung des Anlagenverhaltens in unterschiedlichen Detaillierungsstufen ermöglichen. Langfristig kann darüber hinaus selbst die Integration von CFD (Computational Fluid Dynamics) Simulatoren in Betracht gezogen werden, die eine noch detailliertere Darstellung z.B. des Strömungsverhaltens in Apparaten und Rohren ermöglichen würden [11], wozu allerdings auch die Komponentenmodelle mit den dafür notwendigen Geometrie- und Mediendaten erweitert werden müssten.

4.3 Nutzung von Schnittstellen-Standards

Die Komponentenmodelle müssen von den Modellentwicklern in die Modelldatenbank eingebracht werden, danach werden sie von Modellaggregations-Werkzeugen aus den Datenbanken entnommen und zu (Teil-)Anlagenmodellen zusammengefügt, die dann an die jeweiligen Simulatoren übergeben werden. In jedem Arbeitsschritt ist damit der Austausch von Modellen notwendig. In der prototypischen Realisierung von Model^{CAT} wurden die Simulationsmodelle manuell in die Stammdatenbank von COMOS eingefügt, in diesem internen Format vom Modellaggragationsmodul MAM ausgelesen und zum Teilanlagenmodell zusammengestellt, das dann in den spezifischen Skript-Sprachen an die Simulatoren übergeben wurde. Für eine industrielle Realisierung sollte eine standardisierte und strukturierte Schnittstelle für den Datenaustausch auf der Basis von XML (eXtended Markup Language) genutzt werden, wie sie in den letzten Jahren auch bereits in CAE-Planungssysteme implementiert wurde. Dabei sollte das Auslesen der Modelldaten aus der Datenbank des CAE-Planungssystems mit Hilfe von standardisierten objektorientierten Informationsschnittstellen - z.B. CAEX für die PLS-Modelle [12] und STEP/pdXi für die verfahrenstechnischen Modelle [13] - erleichtert werden.

4.4 Erweiterung des GUI

Das GUI in der prototypischen Realisierung von Model^{CAT} wurde als Zusatzmodul für COMOS in Visual Basic realisiert. Dabei umfasste der Funktionsumfang im wesentlichen die Teilanlagenauswahl im R&I-Fliessbild von COMOS, die Abfrage von Anfangs- und Randbedingungen für die Simulation sowie den Start der Simulatoren. Für die Steuerung der Simulation und Darstellung der Simulationsergebnisse wurden die Ausgabemöglichkeiten der Simulatoren genutzt. In einer industriellen Realisierung sollte eine umfassende Benutzeroberfläche für das Management der virtuellen Inbetriebnahme realisiert werden, die zusätzlich insbesondere die Steuerung der Simulation für verschiedene Inbetriebnahmeszenarien sowie erweiterte Möglichkeiten für die Ausgabe und Auswertung von Simulationsergebnissen ermöglicht. Zwei speziellen Aspekten sollte dabei besonders Rechnung getragen werden:

- **Simulation von Teilanlagen**

In der prototypischen Realisierung von Model^{CAT} wird nur die im R&I-Fliessbild ausgewählte Teilanlage simuliert, die dafür notwendigen Anfangs- und Randbedingungen müssen ermittelt bzw. definiert werden. Gerade bei größeren Anlagen kann es hierbei zu Problemen kommen, da sinnvolle interne Anlagenzustände als Randbedingungen für die Teilanlage u.U. kaum oder nur sehr schwierig zu definieren sind. Dieses Problem könnte vermieden werden, wenn die Anlage in der Regel als Ganzes simuliert wird - die Randbedingungen der Gesamtanlage sind eher bekannt - und später während der Simulation in die interessierenden Teilbereiche der Anlage durch entsprechende Wahl des Anlagenausschnitts mit der passenden Detaillierungsstufe hineingezoomt werden kann.

- **Vollständigkeitsprüfung für Anfangs- und Randbedingungen**

Unabhängig von der oben diskutierten Problematik muss vor jeder Simulation überprüft werden, ob die für die Rechnung des Modells notwendigen Anfangs- und Randbedingungen vollständig definiert sind. Dies wurde bei der prototypischen Realisierung von Model^{CAT} mit Hilfe der Konsistenzprüfung des gewählten VT-Simulators durchgeführt. Bei einer industriellen Realisierung sollte diese Prüfung direkt im GUI implementiert werden, um eine direkte Nutzerführung zur Vervollständigung der notwendigen Randbedingungen zu ermöglichen.

4.5 Einbindung des Inbetriebnehmers und Inbetriebnahmeszenarien

Die prototypische Realisierung von Model^{CAT} hat gezeigt, dass die Grundprozeduren für die Durchführung einer virtuellen Inbetriebnahme - wie die Auswahl der zu simulierenden Teilanlage, die Modellaggregation sowie der Start der Simulation - automatisiert werden können, ohne dass dazu ein Modellierungs- und Simulationsexperte herangezogen wird. In der industriellen Praxis kann die virtuelle Inbetriebnahme vom Anlagenplaner allerdings nur unter Einbeziehung eines Inbetriebnehmer sinnvoll durchgeführt werden, der speziell bei der Definition von realistischen Inbetriebnahmeszenarien (wie z.B. An- und Abfahrphasen, Lastwechsel, Anfahren unterschiedlicher Arbeitspunkte und Störungen) und der Interpretation der Simulationsergebnisse mehr Erfahrung als der Anlagenplaner besitzt und entsprechende Hilfestellungen bei der Analyse des Anlagenverhaltens und bei der Fehlerbehebung geben kann.

5 Schlussbetrachtung und Ausblick

Die Wichtigkeit und Wertschätzung von Methoden zur Vereinfachung der Generierung von Anlagenmodellen wird immer wieder betont, siehe z.B. [14]. Die bisherigen Ansätze erforderten jedoch immer großes Expertenwissen im Bereich Modellierung und Simulation. Mit Model^{CAT} wurde ein Ansatz vorgestellt zur katalogbasierten automatisierten Generierung und Simulation von Anlagenmodellen, womit die Durchführung einer virtuellen Inbetriebnahme durch Anlagenplaner und Inbetriebnehmer ohne den Einsatz von Simulationsexperten in greifbare Nähe rückt.

Können die bisherigen Einschränkungen der prototypischen Realisierung von Model^{CAT} in einer industriellen Realisierung aufgelöst werden durch den Aufbau umfassender Modellkataloge, die Nutzung von standardisierten Austauschformaten für Komponentenmodelle, die Verwendung von professionellen Simulationswerkzeugen sowie durch eine effiziente Benutzerführung mit einem optimierten GUI für das Management der virtuellen Inbetriebnahme, dann kann auch in der industriellen Praxis die virtuelle Inbetriebnahme für den Test der geplanten Anlage vor dem Bau zum Normalfall werden, ohne dass dazu Modellierungs- und Simulationsexperten zur Verfügung stehen müssen. Damit wird es möglich werden, Planungs- und Auslegungsfehler unmittelbar nach Abschluss der Planung an der simulierten Anlage zu erkennen und vor dem Bau zu korrigieren, was zum einen zu einer erhöhten Planungsqualität der Anlage, zum anderen aber auch zu einer vereinfachten und damit kostengünstigeren und kürzeren realen Inbetriebnahmephase führt.

Diese Perspektiven sollten Grund genug sein, die anstehenden Aufgaben bei der industriellen Realisierung einer virtuellen Inbetriebnahmeumgebung anzugehen. Dabei wird der Aufbau einer umfassenden Modellbibliothek für Anlagenkomponenten durch Komponentenhersteller und Modellentwickler eine Schlüsselrolle spielen.

Literatur

[1] SCHUMANN, R. (2007). Virtuelle Inbetriebnahme – Konzept und Realisierungsansätze. VDI-Berichte Nr. 1980, VDI-Verlag, Düsseldorf, 697-706.

[2] HOYER, M., R. SCHUMANN und G.C. PREMIER (2005). Model^{CAT} - ein Konzept zur Modellgenerierung, Simulation und Reglerauslegung für verfahrenstechnische Prozesse. VDI-Berichte Nr. 1883, VDI-Verlag, Düsseldorf, 893-900.

[3] HOYER, M., R. SCHUMANN und G.C. PREMIER (2006). Industrial CACSD for the plant design process. Proceedings of the IEEE conference on computer aided control system design, München, 1843-1848.

[4] INNOTEC GmbH (2007). COMOS. http://www.innotec.de/produktuebersicht.html.

[5] PROCESS SYSTEMS ENTERPRISE Ltd. (2007). London: gPROMS. http://www.psenterprise.com/gproms.

[6] MATHWORKS (2007). Matlab and Simulink. http://www.mathworks.com.

[7] SCHUMANN, R., HOYER, M. und PREMIER, G. C. (2007). Simulation for virtual commissioning. Proceedings of the 6th EUROSIM, Ljubljana, Slovenia, 1-8.

[8] THOMAS, F. (2005). Wie alles begann. Geschichte der Elektronic Design Automation. Design & Elektronik, vol. 9, 100-102.

[9] BARKER, H.A., P.H. HAMMOND, T. HUYNH-QUOC, C.P. JOBLING and D.K. FREDERICK (1991b). A model library for the control engineering community. International IEE Conference Publication 322, Edinburgh, vol. 1, 300-305.

[10] PROCESS SYSTEMS ENTERPRISE Ltd. (2007). London: gO:Simulink. http://www.psenterprise.com/gproms/gosimulink.html.

[11] BEZZO, F., S. MACCHIETTO and C.C. PANTELIDES (2000). A general framework for the integration of computational fluid dynamics and process simulation. Computers & Chemical Engineering, vol. 24, no. 2-6, 653-658.

[12] DRATH R. und M. FEDAI (2004). CAEX – ein neutrales Datenaustauschformat für Anlagendaten. atp 46, Heft 2 (Teil 1) und Heft 3 (Teil 2), R. Oldenbourg Verlag, München.

[13] Britt, H., C.-C. Chen, V. Mahalec und A. McBrien (2004). Modeling and Simulation in 2004: An Industrial Perspective. 6th International Conf. on Foundations of Computer Aided Process Design, CACHE Publications 8.

[14] CAMERON, I.T. (2005). Modelling across the process life cycle: A risk management perspective. In eds. PUIGJANER, L. und A. ESPUNA, ESCAPE-15, Elsevier Science, Barcelona, vol. 20, 3-19.

Analyse und Konzepte zur vereinfachten Virtuellen Inbetriebnahme von Fertigungssystemen

Analysis and concepts for simplified virtual commissioning of manufacturing systems

Dipl.-Ing.(FH) **P. Hoffmann**, Prof. Dr.-Ing. **R. Schumann**
Fachhochschule Hannover, Fakultät II, Hannover, Germany
Dr. **T. M. A. Maksoud** und Dr. **G. C. Premier**
University of Glamorgan, Faculty of Advanced Technology
Pontypridd, Wales, UK

Kurzfassung

Seit mehr als 10 Jahren wird in der Fertigungstechnik über Virtuelle Inbetriebnahme (VIBN) diskutiert, mit deren Hilfe der Test von Fertigungsanlagen simulativ vor dem Bau der Anlage durchgeführt werden soll. Damit könnte der Zeit- und Fehlerbeseitigungsaufwand bei der realen Inbetriebnahme deutlich gesenkt werden, diesem erhofften Vorteil steht allerdings der bis heute als hoch eingestufte Aufwand zur Modellierung und Simulation der Anlage mit allen Steuerungsprogrammen gegenüber. Dies ist gerade für kleine und mittlere Unternehmen (KMU) ein Hinderungsgrund, sich ernsthaft mit der Thematik VIBN zu befassen. In diesem Beitrag sollen vorhandene Ansätze zur Realisierung einer VIBN analysiert und neue Ansätze vorgestellt werden, die im Rahmen eines Forschungsprojektes der University of Glamorgan und der Fachhochschule Hannover entwickelt werden, und mit deren Hilfe eine VIBN mit deutlich geringerem Modellierungsaufwand realisiert werden soll.

Abstract

Virtual commissioning of manufacturing systems has been discussed since years. The expected positive effects for reducing the debugging and correction efforts during real commissioning, however, can only be achieved if detailed manufacturing system models are available. However, the design of such models requires up to now a high level of expertise and considerable efforts which makes virtual commissioning unattractive. This paper describes some new approaches for the systematic and simplified design of manufacturing system models based on model libraries and standardized recipes for the design of component models from CAD data.

1. Einführung

Das industrielle Umfeld, in dem die Entwicklung von Fertigungssystemen stattfindet, zeichnet sich aus durch einen hohen Kostendruck, kürzer werdende Produktlebenszyklen und der Forderung nach schneller Markteinführung. Das führt zu einem immer engeren Zeitrahmen für das (Automatisierungs-)Engineering bei gleichzeitiger Forderung nach geringer Fehlerquote bei Planung und Programmierung.

Fertigungssysteme bestehen u.a. aus Transport- und Handhabungssystemen, Lagersystemen, Bearbeitungsmaschinen, Industrierobotern sowie Steuerungen und Bedien- und Beobachtungssystemen (HMI). Dabei kommen in großem Umfang Standardkomponenten, aber auch anlagenspezifische Konstruktionen zum Einsatz. Die heute noch überwiegend nacheinander abfolgenden Phasen der Entstehung eines Fertigungssystems sind: Anlagenplanung, mechanische Konstruktion, elektrische Konstruktion, Roboter- und SPS-Programmierung, HMI-Programmierung, Aufbau und Inbetriebnahme (Bild 1).

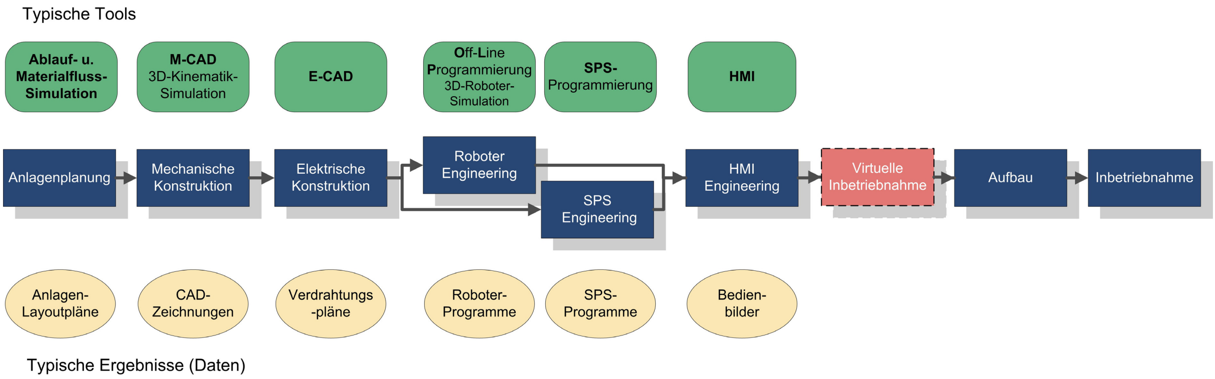


Bild 1: Planungs- und Realisierungsphasen eines Fertigungssystems

Für Anlagenplanung, Konstruktion und Programmierung gibt es eine Vielzahl von auf einzelne Phasen spezialisierten Tools, die auch oft eine Simulationsunterstützung bieten. Heute bestehen jedoch noch große Probleme beim Datenaustausch zwischen den einzelnen Phasen bzw. den eingesetzten Werkzeugen, so dass häufig Mehrfacheingaben nötig sind. Ein Problem bildet dabei das Fehlen eines gemeinsamen allgemein akzeptierten Datenaustauschformats, das jetzt aber mit AutomationML [1], [2] gelöst werden könnte.

Der integrierte Test des Fertigungssystems mit allen Steuerungsprogrammen ist heute in der Regel erst bei der realen Inbetriebnahme möglich, was oft zu teuren

Fehlerbehebungsmaßnahmen und damit zu Zeitverzögerungen führt. So zitieren Zäh und Wünsch [3] eine Studie des VDW (Verein Deutscher Werkzeugmaschinenhersteller) von 1997, nach der bis zu 25% der Gesamtdauer eines Anlagenprojekts auf die Inbetriebnahme fallen. Die Fehlerbehebung der Steuerungssoftware allein beansprucht wiederum 60% der Inbetriebnahmezeit und damit bis zu 15% der gesamten Projektdauer. Als Lösung dieser Probleme empfehlen die Autoren die Durchführung einer Virtuellen Inbetriebnahme (VIBN) vor dem Aufbau der Anlage.

2. Virtuelle Inbetriebnahme – eine kurze Zwischenbilanz

Bei einer VIBN wird mit Hilfe eines Simulationsmodells der Fertigungsanlage die Inbetriebnahme simulativ vor dem Aufbau der Anlage (Bild 1) durchgeführt. Das Ziel ist dabei, Planungs- und Auslegungsfehler frühzeitig zu erkennen und bereits in den Planungsdaten zu korrigieren, anstatt erst nach dem Aufbau der Anlage bei der Inbetriebnahme erkannte Fehler durch Umbauten zu beheben. Zur Überprüfung der Geometrie und der mechanischen Auslegung genügt dazu eine 3-D-Simulation des zu erwartenden mechanischen Sollverhaltens der Anlage, zur Prüfung von Steuerungsprogrammen eine Anlagensimulation, welche das Sollverhalten an den Ein- und Ausgängen der Steuerungen widerspiegelt (Soll-Simulationen). Eine VIBN, die auf der Basis solcher Soll-Simulationen durchgeführt wird, ist entsprechend auf die Aufdeckung mechanischer und geometrischer Auslegungsfehler einerseits und der Steuerungsfunktionen andererseits beschränkt. Soll zusätzlich auch der Einfluss der realen Steuerungsprogramme auf das mechanische Verhalten im Detail geprüft werden, so ist dazu eine Modellierung und Simulation der gesamten Wirkungsketten von den Steuerungen über Aktoren und Stellantriebe bis hin zu den Bewegungen der Fertigungsanlage notwendig. Eine derartige Simulation, die das zu erwartende Verhalten der realen Anlage mit allen Wirkungszusammenhängen ganzheitlich und realitätsnah abbildet, wird hier als Ist-Simulation bezeichnet und ermöglicht bei der VIBN die Aufdeckung auch von Fehlern in den Wirkungsketten aller Funktionen von den Steuerungen bis hin zu den mechanischen Bewegungen.

In welchem Umfang eine VIBN die bei der realen Inbetriebnahme zu erwartende Fehlerquote verringern kann, zeigt eine in [4] beschriebene Studie zur Wirtschaftlichkeit einer VIBN. Dabei wurde ein Experiment mit zwei Gruppen von Programmierern zu je 30 Personen durchgeführt, die jeweils ein Steuerungsprogramm für eine SPS-gesteuerte Dosenpresse mit 10 Ausgängen für Aktoren und 17 Eingänge für Sensoren entwerfen sollten. Die erste

Gruppe entwickelte das Steuerungsprogramm in herkömmlicher Weise und nahm es direkt an der realen Anlage in Betrieb, die zweite testete ihr Programm mit Hilfe eines simulierten Anlagenmodells und nahm die reale Anlage erst nach erfolgreicher VIBN in Betrieb. Die Ergebnisse zeigten, dass mit der zweiten Vorgehensweise rund 75% der realen Inbetriebnahmezeit eingespart werden konnten, was auf die deutlich geringere Fehlerquote in dem mit der VIBN vorab überprüften Programm zurückgeführt wurde. Dieses Experiment zeigte deutlich die Vorteile einer VIBN, allerdings wurde hier der Aufwand für die Modellierung nicht explizit mit in Betracht gezogen, da das virtuelle Maschinenmodell bereits im Vorfeld der Studie entwickelt wurde. Die Autoren betonen jedoch die Notwendigkeit einer vereinfachten bzw. beschleunigten Modellentwicklung z.B. durch Standardisierung von Komponenten und Anlagenteilen und eine automatische Modellgenerierung.

3. Realisierungsansätze für eine VIBN

Die Möglichkeiten, eine VIBN durchzuführen, werden seit einigen Jahren von Industrie und Hochschulen untersucht. Ein früher Ansatz zu einer Virtuellen Inbetriebnahme findet sich in [5]: Als Anlagenbeispiel wurde ein Palettentransfersystem gewählt, das im Simulationstool ARENA modelliert und mit einer realen Steuerung gekoppelt wurde. In verschiedenen Experimenten konnten mit diesem Testaufbau Unterschiede zwischen dem vorher festgelegten Soll-Verhalten und dem per VIBN überprüften simulierten Ist-Verhalten entdeckt werden, z.B. Fehler durch übersehene inverse Logik von Stoppern oder auch Probleme im Materialfluss, die ohne VIBN nur schwer vor der realen Inbetriebnahme zu finden gewesen wären. Der Modellierungsaufwand wurde (wohl aus gutem Grund) nicht kommentiert. Nach [5] kann der Test einer Steuerung an einer geplanten Anlage in 4 Grundkonfigurationen durchgeführt werden:

1) Reale Steuerung an realer Anlage: klassische Testkonfiguration für die reale Inbetriebnahme.
2) Reale Steuerung an simulierter Anlage: in [5] als „Soft-commissioning“, häufig auch als „Hardware-in-the-loop“ (HIL) bezeichnet. Benötigt vorab die reale Steuerung, ermöglicht aber eine VIBN vor dem Aufbau der Anlage.
3) Simulierte Steuerung an realer Anlage. In [5] als „Reality in the loop“ bezeichnet.
4) Simulierte Steuerung an simulierter Anlage: ermöglicht eine komplette VIBN.

Eine VIBN in der Konfiguration 2) oder 4) erfordert die Kopplung zwischen der realen/ simulierten Steuerung und der Anlagensimulation, die z.B. über OPC realisiert werden kann. In der 4. Konfiguration kann die Simulation von Steuerung und Anlage auch in einem Tool erfolgen, wie dies z.B. für S7-Steuerungen und diverse Robotersteuerungen im Anlagensimulationstool CIROS [6] möglich ist.

In der Literatur gibt es schwerpunktmäßig zwei Bereiche, die sich mit der VIBN für Fertigungsanlagen befassen. Der erste Bereich beschäftigt sich mit dem Test und der Verifikation von Steuerungsprogrammen für Fertigungsanlagen. Ein Steuerungsprogramm kann auf sehr unterschiedliche Arten geprüft werden, vom Durchspielen von Testszenarien an der realen Anlage oder per Simulation bis hin zur formalen Logikanalyse. Eine umfangreiche Übersicht der verschiedenen Testmöglichkeiten (mit 18 Methoden) findet sich in [7]. Die hier dargestellten Möglichkeiten reichen von verschiedenen Simulationsmethoden bis hin zu den formalen Logikanalysemethoden. Zusätzlich zu den vier oben genannten Grundkonfigurationen werden noch ergänzend Hardware-Testpanel zur Prozessdarstellung genannt, die aber nur eine grobe Simulation der Anlage erlauben, oder die Nachbildung des Prozessmodells innerhalb des Steuerungssystems durch Simulationsbausteine.
Eine gute Übersicht über die zahlreichen Arbeiten zur Verifikation von Steuerungsprogrammen mit Hilfe von formalen Analysemethoden findet sich in [8]. Das formalisierte Programm soll dabei bezüglich einer vorgegebenen Spezifikation des Soll-Verhaltens allein oder zusammen mit einem abstrakten Anlagenmodell geprüft werden. Ein derzeit noch nicht zufriedenstellend gelöstes Problem ist die Erstellung von hinreichend genauen abstrakten Anlagenmodellen in der Sprache des Analysetools, eine industriell nutzbare Umsetzung für die VIBN ist heute noch nicht absehbar.

Der zweite Bereich befasst sich mit den Möglichkeiten der Digitalen Fabrik [9], [10] zur Realisierung einer VIBN. Dabei wird versucht, mittels des Zusammenwirkens von digitalen Modellen, rechnergestützten Planungs- und Auslegungsmethoden und den dazu notwendigen Software-Tools unter Zuhilfenahme eines integrierten Datenmanagements eine durchgängige Planung, Simulation und Validierung von Fertigungsprozessen und –systemen zu ermöglichen [11]. Die Durchführung soll sich dabei auf alle Phasen des Entstehungsprozesses eines Fertigungssystems erstrecken. In einigen Phasen gibt es schon eine intensive Nutzung von Simulationen, z.B. Ablauf- und Materialflusssimulationen in der Anlagenplanung, 3D-Kinematiksimulationen in der mechanischen Konstruktion bis hin zum Zusammenspiel mit Produktdaten beim Digital Mock-Up (DMU). Offline-Programmierung (OLP) und 3-D Robotersimulation werden seit vielen Jahren genutzt und haben inzwischen einen hohen Reifegrad erreicht. Vereinzelt werden inzwischen auch Ergonomiesimulationen mit 3-D Mensch-Modellen genutzt. Der Schwerpunkt der bisher genutzten Simulationen liegt jedoch im Planungsbereich und dabei hauptsächlich in der Absicherung des Soll-Verhaltens der mechanischen Anlagenkomponenten.

Die im Kontext der Digitalen Fabrik bestehende Möglichkeit, für die VIBN die komplexen geschlossenen Toolsuiten der großen Anbieter zu nutzen, führt zu hohem Schulungsaufwand bei den Planungs- und Inbetriebnahmeingenieuren und dem Vor-Ort-Einsatz von Spezialisten der Tool-Anbieter und wird daher derzeit praktisch nur von Großunternehmen z.B. aus der Automobilindustrie, und dort wegen des hohen Aufwandes auch nur punktuell, genutzt. Kleine und mittlere Unternehmen (KMU) können sich diesen großen Aufwand in der Regel nicht leisten. Insbesondere die Erstellung von Simulationsmodellen unter Berücksichtigung der Mechanik- und Elektroplanungsdaten sowie der Steuerungs- bzw. Roboterprogramme ist mit hohem Aufwand verbunden.

Bereits diese kleine Übersicht zeigt die heutigen Möglichkeiten und Grenzen für die Realisierung einer VIBN für fertigungstechnische Anlagen auf: Einerseits werden stets die positiven Effekte einer VIBN wie die Verkürzung der realen Inbetriebnahmezeit oder die Verbesserung der Planungsqualität herausgestellt [3 bis 5], andererseits wird aber auch immer wieder auf die Schwierigkeiten und den großen Aufwand hingewiesen, der nötig ist, um aussagefähige Anlagenmodelle für die Simulation zu erzeugen [12], [13]. Dieser Aufwand ist bis heute das Haupthindernis für die Durchführung einer VIBN. Damit kommt der Entwicklung von systematischen Methoden zur vereinfachten Erzeugung von Anlagenmodellen bei der Realisierung einer VIBN eine entscheidende Bedeutung zu.

4. Generierung von Anlagenmodellen für die VIBN

Zur Untersuchung der Möglichkeiten zur vereinfachten Generierung und Nutzung von Anlagenmodellen für die VIBN wurde von der University of Glamorgan und der Fachhochschule Hannover ein gemeinsames Projekt initiiert.

Ausgangspunkt für dieses Projekt ist das 3D-Anlagensimulations-Tool CIROS [6], das am Institut für Roboterforschung (IRF) der Universität Dortmund ursprünglich als Robotersimulationssystem COSIMIR [14], [15] entwickelt wurde. CIROS ermöglicht die einfache Zusammenstellung und Simulation von Fertigungsumgebungen auf der Grundlage einer Modellbibliothek, wobei die darin enthaltenen Modelle bereits die Wirkungsketten von den Sensoren und Aktoren bis hin zu den mechanischen Bewegungen komplett abbilden. So lässt sich der Einfluss der Steuerungs- oder Roboterprogramme, die in CIROS auf emulierten Steuerung ablaufen, über die gesamte Wirkungskette bis hin zu den damit veranlassten mechanischen Bewegungen verfolgen und testen, gleichzeitig kann mit der 3D-

Simulation auch die Verträglichkeit dieser Bewegungen mit der Umgebung illustriert und überprüft werden. Solange das Modell der Fertigungsanlage aus den Bibliotheksmodellen (High-Level-Modellierung) zusammengestellt werden kann, lässt sich eine VIBN relativ einfach aufsetzen, wobei der Zusatzaufwand die Zusammenstellung des Anlagenmodells im Editor, die Herstellung der Ein/Ausgangsverbindungen sowie die Übertragung des Steuerungsprogramms auf die emulierte Steuerung umfasst.

Sollen allerdings neue Komponenten mit simuliert werden, so müssen neue Bibliotheksmodelle auf der Basis der CAD-Daten mit erheblichem Aufwand generiert werden. Diese mehrstufige Low-Level-Modellierung umfasst die gesamte Wirkungskette von der Steuerung bis zu den mechanischen Bewegungen und erfordert entsprechend eine erhebliche Expertise zur Erstellung der Komponentenmodelle. Der hohe Aufwand einer derartigen geometrischen, funktionalen und elektrischen Low-Level-Modellierung (notwendig für CIROS, aber auch in ähnlicher Weise für andere Simulationstools wie WinMOD oder DELMIA Automation V5) wird in der folgenden Beschreibung verdeutlicht:
Geometrische Modellierung: Erzeugung eines strukturierten CAD-Modells.
Ausgangspunkt der geometrischen Modellierung ist ein 3D-CAD Modell der Komponente, das entweder selbst erstellt werden muss oder vom Komponentenhersteller stammt. Das Simulationssystem verlangt eine Hierarchisierung der 3-D-Objekte und Strukturierung in bewegliche Elemente, was bei herkömmlichen CAD-Daten oft nicht der Fall ist und dann umfangreiche manuelle Nacharbeiten notwendig macht. Bei zu komplexen bzw. zu detaillierten CAD-Daten wird häufig noch eine manuelle Vereinfachung notwendig.
Funktionale Modellierung: Erstellung eines strukturierten mechanisch-funktionalen Modells.
Hier müssen den geometrischen (Sub-)Modellen aktorische Funktionen (mechanische Translation, Rotation, Greifmechanismen …) und sensorischen Funktionen (induktiv, optisch, …) manuell zugeordnet werden. Dabei entsteht ein strukturiertes Funktionsmodell mit Unterfunktionen wie Zylinder, Drehtisch, Sensor, Greifer u.s.w.
Elektrische Modellierung: Erstellung des strukturierten mechatronischen Gesamtmodells.
In der abschließenden elektrischen Modellierung werden den Objekten des mechanisch-funktionalen Modells manuell elektrische Ein- und Ausgänge zugeordnet, die später mit den Steuerungsein/ausgängen verbunden werden können.

Es existieren keine Modellierungsrichtlinien für die Low-Level-Modellierung, jeder Modellierungsexperte hat seinen eigenen Stil – was bedeutet, dass dieselbe Komponente von zwei Experten unterschiedlich modelliert werden kann. Nach der in [16] genannten

Zielsetzung ist die Low-Level-Modellierung für den Einsatz beim Komponentenhersteller gedacht. Dort sollen technisch versierte Modellierer mit Detailkenntnissen der mechanischen, elektrischen, pneumatischen oder geometrischen Daten der Komponenten Simulationsmodelle erstellen, testen und als Bibliotheken zur Verfügung stellen. Diese Modellbibliotheken sollen dann von den Anlagenplanern im Rahmen der High-Level-Modellierung bei der Zusammenstellung der virtuellen Anlage verwendet werden.

5. Neue Konzepte zur vereinfachten VIBN von Fertigungsanlagen

Für verfahrenstechnische Anlagen wurde im Rahmen des Model^{CAT}-Projekts [17] an der Fachhochschule Hannover demonstriert, dass es prinzipiell möglich ist, auf Basis von Planungsdaten eines CAE-Planungssystems auf der Basis von Modellbibliotheken automatisch Simulationsmodelle zu generieren, mit deren Hilfe eine VIBN möglich wird. Spätestens am Ende der Anlagenplanung mit abgeschlossener Programmierung sollte es so sehr schnell möglich werden, eine VIBN durchzuführen, da dann ja auch alle Daten zu Aufbau und Inbetriebnahme der realen Anlage im Planungssystem vorliegen. Im aktuellen Projekt soll untersucht werden, inwieweit der Model^{CAT}-Ansatz auf die Fertigungstechnik übertragbar ist. Dazu soll ein CAE-Planungsystem zur Erfassung aller Planungsdaten eingesetzt werden, die in einer objektorientierten Datenbank abgelegt werden. Dazu gehören auch die mechatronischen Objekte und die Steuerungsprogramme der Fertigungsanlage. Kombiniert man nun das CAE-Planungssystem mit einem Anlagensimulator wie CIROS, so werden im Idealfall die mechatronischen Komponenten bereits im CAE-Planungssystem zusammengefügt und danach mit Hilfe passender Bibliotheksmodelle in CIROS simuliert. Dabei könnte die manuelle High-Level-Modellierung des Simulationsmodells in CIROS zumindest teilweise entfallen, das Simulationsmodell der Anlage könnte mit den Anordnungsinformationen aus dem CAE-Planungssystem weitgehend automatisiert erstellt werden.

Zur Überprüfung dieses Konzeptes zur vereinfachten Generierung von Simulationsmodellen für eine VIBN wurde die prototypische Engineering-, Modellierungs- und Simulationsumgebung (PEMS, Bild 2) konzipiert, in der CIROS als zentrales Simulationswerkzeug eine Schlüsselrolle spielt. Kombiniert wird CIROS mit dem CAE-Planungssystem Comos [18] mit objekt-orientierter Datenbank. Mit PEMS soll zunächst untersucht werden, wie aus den CAE-Planungsdaten über ein Zusatzwerkzeug automatisiert CIROS-Simulationsmodelle generiert werden können, sofern die Fertigungsanlage aus

Standardkomponenten zusammengefügt wird, für die CIROS-Bibliotheksmodelle vorhanden sind (High-Level-Modellierung).

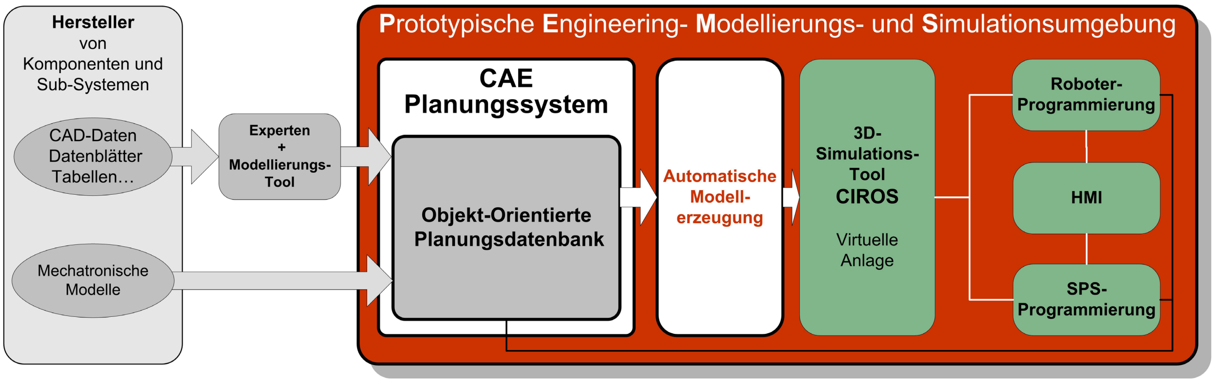


Bild 2: Prototypische Engineering- Modellierungs- und Simulationsumgebung (PEMS)

Im Rahmen des Projektes sollen dann auch Wege zur systematischen Ergänzung und Erweiterung der Modellbibliothek untersucht werden, da bei den meisten realen Projekten davon ausgegangen werden muss, dass neben Bibliothekskomponenten auch spezielle und neue Komponenten zum Aufbau der Fertigungsanlage eingesetzt werden. Dazu wird im Rahmen des Projektes untersucht, inwieweit auch die Low-Level-Modellierung durch eine Systematisierung der Modellierungsschritte vereinfacht werden kann. Einen Schlüssel dazu bilden Anforderungen an die Konstrukteure in Bezug auf die funktionsorientierte Strukturierung der CAD-Daten. Darüber hinaus wird untersucht, wie eine systematische Sammlung von Komponentenmodellen, die z.B. bei Komponentenherstellern oder Hochschulen entwickelt werden, organisiert werden kann. Diese Komponentenmodelle sollten in einer neutralen Form abgelegt und ausgetauscht werden, wozu die Vereinbarung von standardisierten Strukturen und Datenformaten für mechatronische Simulationsmodelle nötig sein wird. Einen guten Ansatz dazu bildet AutomationML [1], [2], das sich zum gemeinsamen allgemein akzeptierten Datenaustauschformat auch für solche Modelle entwickeln könnte.

5.1 Überprüfung der neuen Konzepte in der Testumgebung RFM

Als Testumgebung für die Anwendung der prototypischen Engineering- Modellierungs- und Simulationsumgebung (PEMS) dient eine kleine fertigungstechnische Zelle der Fachhochschule Hannover, das „Robotergestützte Flexible Montagesystem“ (RFM). Dieses besteht aus einem von einer Siemens S7-300 SPS gesteuerten Transportsystem mit autonomen, spurgebundenen Transportwagen, einem dazugehörigen Aufzug sowie zwei

Roboter-Montagezellen, ausgerüstet mit einem SCARA-Roboter und einem Portalroboter (Bild3). Der Portal-Roboter ist mit einem über CoDeSys [19] programmierbaren Berger-Lahr Motion Controller ausgerüstet.

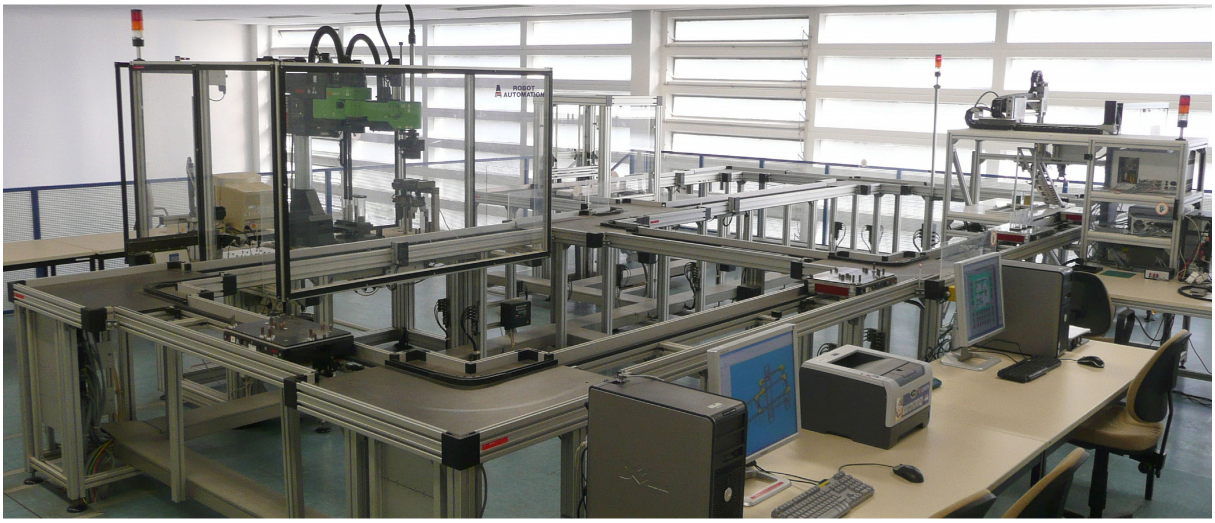


Bild 3: Testumgebung „Robotergestütztes Flexibles Montagesystem“ (RFM).

Zur Überprüfung der Vorgehensweise zur Durchführung einer vereinfachten VIBN wird der Planungsprozess der Fertigungszelle mit der PEMS exemplarisch durchgeführt. Dabei soll gezeigt werden, wie aus den Planungsdaten in Comos das Anlagensimulationsmodell weitgehend automatisiert generiert und mit CIROS simuliert werden kann. Als Voraussetzung werden dazu Simulationsmodelle der Anlagenkomponenten für die CIROS-Modellbibliothek erstellt, wobei unterschiedliche Vorgehensmodelle für die Low-Level-Modellierung sowohl mit Experten als auch mit unerfahrenen Modellierern hinsichtlich Modellgüte und Modellierungsaufwand getestet werden.

6. Zusammenfassung und Ausblick

Das Projekt an der Fachhochschule Hannover soll einen Weg aufzeigen, wie Automatisierungs- und Inbetriebnahme-Ingenieure grundsätzlich in die Lage versetzt werden können, die Simulation eines Fertigungssystems auf der Grundlage von Planungsdaten durchzuführen und im Rahmen einer VIBN zu nutzen, ohne selbst Spezialisten für Modellierung und Simulation zu sein. Die dazu notwendigen Voraussetzungen wie der Aufbau einer objekt-orientierten Bibliothek vom Simulationsmodellen von Anlagen-komponenten, die später vornehmlich von den Komponenten-Zulieferern zu füllen sein wird, sowie eine verbesserte Unterstützung durch die einzusetzende Werkzeugumgebung, werden im Laufe des Projektes Gegenstand von ausführlichen Untersuchungen und Tests sein.

Aus Sicht der Autoren ist es sinnvoll und notwendig, den Aufwand der Modellierung so weit wie möglich nach vorne zu verlagern, also zu den Herstellern von Komponenten und Sub-Systemen. Diese sollten in die Lage versetzt werden, zusätzlich zu den heute verfügbaren CAD-Daten für ihre Hardwarekomponenten auch mechatronische Simulationsmodelle anzubieten, die mit Hilfe von standardisierten Datenformaten in verschiedene Simulationsumgebungen importierbar sind. Wenn die Hersteller nicht mehr wie heute nur CAD-Daten (die heute für die Simulation direkt nur schwer verwendbar sind) und (elektronische) Datenblätter anbieten, sondern auch simulierbare mechatronische Komponentenmodelle, könnte der Modellierungs-Aufwand für die Durchführung einer VIBN beim Anlagenplaner und Anwender entfallen oder zumindest deutlich reduziert werden. Um dieses Ziel zu erreichen, sollten die Komponentenhersteller durch vereinfachte Vorgehensmodelle für die Low-Level-Modellierung unterstützt werden, wie sie im Rahmen des vorgestellten Projektes entwickelt werden. Diese Komponentenmodelle könnten in Modellbibliotheken abgelegt und dann in CAE-Planungsumgebungen wie dem skizzierten PEMS importiert werden. Nach Abschluss der Planung sollte damit aus den Planungsdaten das Simulationsmodell für die VIBN weitgehend automatisiert generiert werden, wodurch Anlagenplaner und Inbetriebnehmer mit geringem Zusatzaufwand in die Lage versetzt werden könnten, nach Abschluss der Planung und vor dem Aufbau der Anlage eine virtuelle Inbetriebnahme durchzuführen. Das vorgestellte Projekt soll dazu einen Beitrag liefern, indem es anhand eines Fallbeispieles Lösungsansätze für die vereinfachte Realisierung einer VIBN aufzeigt.

Literatur

[1] Drath, R., Lüder, A., Peschke, J., und Hundt, L.: "AutomationML - the glue for seamless automation engineering", *IEEE International Conference on Emerging Technologies and Factory Automation, 2008. ETFA 2008*, Hamburg, 2008, S. 616-623

[2] Grimm, B., Hundt, L., Lüder, A., und Peschke, J.: "Universelles Datenaustauschformat", *A&D Kompendium 2008/2009*, 2008, S. 266-268

[3] Zäh, M. F. und Wünsch, G.: "Schnelle Inbetriebnahme von Produktionssystemen", *wt Werkstattstechnik online,* 95, 9, 2005, S. 699-704

[4] Zäh, M. F., Wünsch, G., Hensel, T. und Lindworsky, A.: "Nutzen der virtuellen Inbetriebnahme: Ein experiment - Use of virtual commissioning: An experiment", *ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb,* 101, 2006, S. 595-599

[5] Auinger, F., Vorderwinkler, M., und Buchtela, G.: "Interface driven domain-independent modeling architecture for 'soft-commissioning' and 'reality in the loop'", *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, USA, 1999, S. 798-805

[6] RIF - Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e. V.: "*CIROS Engineering*", http://www.ciros-engineering.com

[7] Danielsson, F., Moore, P., und Eriksson, P.: "Validation, off-line programming and optimisation of industrial control logic", *Mechatronics,* 13, 2003, S. 571-585

[8] Bani Younis, M. und Frey, G.: "Formalization of existing PLC programs: A Survey", *CESA 2003*, Lille (France), 2003

[9] Kühn, W.: "Digital Factory - Simulation Enhancing the Product and Production Engineering Process", *Proceedings of the Winter Simulation Conference WSC 06*, 2006, S. 1899-1906

[10] Schlögl, W.: "Einsatz der Digitalen Fabrik von der Anlagenplanung bis in den Laufenden Betrieb", *GMA-Kongress 2007*, Baden-Baden, 2007, S. 717-725

[11] VDI: "VDI-Richtlinie 4499 - Blatt 1: Digitale Fabrik - Grundlagen ", VDI-Gesellschaft Fördertechnik Materialfluss Logistik, Düsseldorf, 2008

[12] Moore, P. R., Pu, J., Ng, H. C., Wong, C. B., Chong, S. K., Chen, X., Adolfsson, J., Olofsgard, P., und Lundgren, J. O.: "Virtual engineering: an integrated approach to agile manufacturing machinery design and control", *Mechatronics,* 13, 2003, S. 1105-1121

[13] Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., Han, K. H., und Chang, M.: "Development of Virtual Simulator for Visual Validation of PLC Program", *International Conference on Computational Intelligence for Modelling Control and Automation,and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*

[14] Freund, E., Rossmann, J., Uthoff, J., und can der Valk, U.: "Towards realistic simulation of robotic workcells", *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94.* , Munich, 1994, S. 39-46

[15] Freund, E. und Rossmann, J.: "Systems approach to robotics and automation", *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Nagoya 1995, S. 3-14

[16] Wischnewski, R. und Freund, E.: "COSIMIR Transport: Modeling, simulation and emulation of modular carrier based transport systems", *International Conference on Robotics & Automation*, New Orleans, LA, United States, 2004, S. 5171-5176

[17] Hoyer, M., Schumann, R., Hoffmann, P. und Premier, G. C.: "Virtuelle Inbetriebnahme mit Model^{CAT} - Vom Prototypen zum industriellen Einsatz / Virtual Start-Up with Model^{CAT} - From Prototypical Realisation to Industrial Implementation", *Automation 2008*, Baden-Baden, 2008, S. 203-206

[18] Comos Industry Solutions: "*COMOS*", http://www.innotec.de

[19] 3S - Smart Software Solutions: "*CoDeSys*", http://www.3s-software.com

VIRTUAL COMMISSIONING OF MANUFACTURING SYSTEMS A REVIEW AND NEW APPROACHES FOR SIMPLIFICATION

Peter Hoffmann
Reimar Schumann
Fakultät II – Abt. Maschinenbau
University of Appl. Sc. and Arts Hannover
Ricklinger Stadtweg 120
30459 Hannover, Germany
E-mail: peter.hoffmann@fh-hannover.de

Talal M.A. Maksoud
Giuliano C. Premier
Faculty of Advanced Technology
University of Glamorgan
8 Forest Grove
Pontypridd, Wales, UK, CF37 1DL
E-mail: tmaksoud@glam.ac.uk

KEYWORDS
Modelling, virtual commissioning, manufacturing systems.

ABSTRACT

Virtual commissioning (VC) of manufacturing systems has been researched for more than 10 years. Its intention is to test manufacturing systems and associated control programs through simulation conducted before the real systems are realised. The expected benefits in reducing debugging and correction efforts expended during real commissioning, however, can only be achieved if sufficiently detailed manufacturing system models are available for simulation. To date, the design of such models has certainly required a high level of expertise and considerable effort, which makes virtual commissioning unattractive, especially for small and medium-sized enterprises (SME). After reviewing the current status of VC, this paper describes some new concepts for the systematic and simplified design of manufacturing system models for VC based on model libraries and standardized recipes for the design of component models from CAD data. This work is carried out as part of the research cooperation between the University of Glamorgan and the University of Applied Sciences and Arts Hannover.

INTRODUCTION

Today the design of manufacturing systems takes place in an industrial environment characterized by significant cost constraints, shortening of product life-cycles and strategies for rapid time-to-market. For these reasons, the timeframe for manufacturing system engineering is progressively tightening whereas the demands on planning accuracy and planning quality are growing.

Manufacturing systems consist of different elements such as storage, magazines, conveyors, handling and transportation systems, machining and assembling tools, robots, control and HMI systems, often in a combination of a large number of standard parts and some purpose-built parts or sub-systems. The development of a manufacturing system, in general, comprises several phases: facility design, mechanical engineering, electrical engineering and automation engineering (programming of robots, PLCs and HMI), which are often sequentially executed.

There are many different powerful and specialized tools for design and engineering, often with integrated simulation, however, there are great problems regarding data exchange between the different engineering phases and the different associated tools. A typical problem is the repeated data entry generating random errors. One principal problem is the missing generally accepted data exchange format which might be solved by AutomationML® (Drath et al. 2008a).

Up to now, after completion of engineering, procurement and assembly, the real commissioning is finally done. Conventionally an integrated test of the planned manufacturing system cannot be done before it has been built; consequently a considerable number of design problems and faults often remains undetected before the first system start-up. This leads in general to time and money consuming corrective measures being required during commissioning and the early production phases resulting in time delays and increased costs to all parties involved.

According to (Zäh and Wünsch 2005), referencing a study of the VDW (German Association of machine tool builders), the commissioning time consumes up to 25% of the time available for plant engineering and construction; and up to 15% is expended on correcting errors in the control software alone. As a possible solution to these problems the authors propose virtual commissioning (VC).

During VC, a simulation model of the manufacturing system is used to allow commissioning through simulation, before building the system. The goal is the early detection and correction of errors generated during planning, design and programming. VC may be conducted during all engineering phases, but at the latest after engineering has been completed.

An experimental study in virtual commissioning (Zäh et al. 2006) shows the positive effects of VC on the error

Proceedings 24th European Conference on Modelling and
Simulation ©ECMS Andrzej Bargiela, Sayed Azam Ali
David Crowley, Eugène J.H. Kerckhoffs (Editors)
ISBN: 978-0-9564944-0-5 / ISBN: 978-0-9564944-1-2 (CD)

rate during real commissioning. The study was conducted with two groups of control programmers. Each group had 30 individuals. One group applied VC to the software development for a machine. The results were compared to those from the second group of programmers that did not use VC. A tin can moulding press was used as a test bed, with a Siemens S7-300 PLC which uses 10 actuator outputs and 17 sensor inputs. One group programmed the PLC and tested the program afterwards in a real world commissioning on a real machine. The other group programmed by using a virtual machine model. They did not execute the real commissioning before achieving successful VC. The results showed a reduction of real commissioning time by 75%, resulting from enhanced software quality at the start of real commissioning. This emphasizes the advantages of running a VC, but the virtual machine model had already been developed in the run-up to this study, and this effort was not taken into consideration. In their conclusion the authors point out the need for simplified and accelerated model building.

APPROACHES TO REALISING VC

Investigating the feasibility of VC has been an academic and industrial research objective for several years. For separate verification of geometry, kinematics and mechanical design, a 3D simulation of the expected and specified mechanical behaviour is sufficient. A VC based on such simulations is able to detect mechanical resp. geometrical planning errors. For separate verification of the control programs, a simulation reflecting the specified behaviour of the manufacturing system mechanics at I/O level is needed. A VC based on such simulations is able to detect deviations from the specified control functions. If the impact of control programs on the 3D mechanical behaviour of the manufacturing system is to be tested in detail in an integrated manner, modelling and simulation of the complete functional chains from control programs through sensors, actuators and drives onto the mechanical movements, is necessary which includes both, simulation of mechanical behaviour and of control programs .

An early approach to VC was presented by Auinger et al. (1999) and termed as “soft-commissioning”. The authors propose VC because of the time consuming and expensive testing and debugging required by PLC based control software. As a test bed for their VC approach, the authors used a PLC controlled pallet transfer system. The modelling of the transfer system, based on the simulation tool ARENA®, was done for the design and optimization of the manufacturing process. This model contained all of the control logic. Later they reduced the model, and the model used for the soft-commissioning no longer contained the control logic. The model provided only realistic I/O signals for the coupled PLC and visual feedback. In their experiments, they found malfunctions of stoppers caused by overlooked inverse logic and problems in the material flow. Normally such errors would only be detected during real commissioning. The modelling effort was not (probably with good cause) discussed.

In accordance with Auinger et al. (1999) the verification of control can be arranged in four basic system configurations:

1. Real plant and real control system: The traditional way of testing during real commissioning.
2. Simulated plant and real control system: “Soft-commissioning” often called “hardware in the loop” (HIL). The hardware controller is necessary in advance, but a VC before building the plant is possible.
3. Real plant and simulated control system: termed “Reality in the loop” by the authors.
4. Simulated plant and simulated control system: This offers a complete VC.

A VC in the second or fourth configuration requires the coupling between real or simulated controller and the mechanical plant simulator which can be realized with e.g. OPC. In the fourth configuration the simulation of controller and plant can run in one tool. This is for example possible for several robot controller and Siemens S7 PLC inside the plant simulation tool CIROS® (RIF 2010).

Currently, two research domains are linked to VC. The first domain deals with test and verification of control programs. This is possible by applying many different techniques, from testing on the real plant, to formal logic analysis. A comprehensive survey of 18 different methods can be found in (Danielsson et al. 2003). In addition to the 4 basic configurations mentioned above, the authors refer to hardware test panels for process simulation, which allow only a rough plant simulation, or simulation with process models blocks inside the control software tool. A useful survey of the research area related to formalization of existing PLC code was given by (Bani Younis and Frey 2003). For VC the verification of formalized control code against an abstract model of the plant, in the language of the analytical tool, would be necessary. The generation of such models is a theoretical and practical problem and a usable industrial implementation for VC is not foreseeable today.

In (Thapa et al. 2006) an approach for offline verification and validation of the control logic is defined and justified by the lack of tools to provide completely integrated solutions for the verification and validation of control logics. The authors propose a III-phase method and classify these phases as: Manual testing, Model checking and VC. Manual testing means checking the code on a softPLC (or simulated PLC like S7-PLCSIM) by user inputs; this is only useful for small programs or parts of programs. Model checking uses formalization as described by (Bani Younis and Frey 2003). The standard IEC 61131-3 code is converted to an intermediate language, transformed to timed automata and the model checked. For the virtual commissioning they coupled a virtual plant model and a softPLC running the checked standard code. The III-phase method requires considerable effort and expertise to build the virtual system, which is costly. The authors referred to the market situation where customers want to validate the process using VC or simulation using 3-D models and not textual verification only.

The second domain addresses the scope of the "Digital Factory" (Kühn 2006, Schlögl 2007). By means of digital models, computer-aided planning and design, computer-aided engineering, associated software tools and with the aid of integrated data management, the "Digital Factory" would permit integrated planning, simulation and validation of manufacturing processes and systems (VDI 2008). The use of the "Digital Factory" should extend to all phases of manufacturing system development. Nowadays simulation is already used intensively in some phases, e.g. material flow simulation for facility design, 3D simulation of kinematics for mechanical engineering and possibly interaction with product data at Digital Mock-Up (DMU). Off-line programming and 3D robot simulation have been used for many years, sometimes 3D human models are used today to evaluate ergonomics in design. Up to now the focus of simulation has been primarily in design and mechanical engineering, where fit and specified behaviour of mechanical plant components must be ensured.

In the context of the "Digital Factory" it is in principle possible to use the complex off the shelf engineering tools from major vendors for a VC, but this usually requires a high level of training or in-house secondment of specialists from the vendor so that only large enterprises (e.g. in the automotive industry) selectively choose to conduct a VC. These "Digital Factory" solutions cannot solve the problems of small and medium-sized enterprises, because they normally do not have the resources to use these techniques (Westkämper et al. 2003), and there is therefore limited use of the "Digital Factory" today (Drath et al. 2008b). The generation of simulation models with consideration of mechanical and electrical planning data and control programs is especially associated with high effort.

In (Wischnewski 2007) the author described the VC procedure when using the simulation tool COSIMIR® and its extension module Cosimir Transport (Wischnewski and Freund 2004) for carrier based transport systems. The author justifies the VC by the exceedingly error-prone control design of transport systems originating from the use of many sensors and actuators and the complex program sequences required for the routing strategies implemented. Rossmann et al. (2007) described a VC with COSIMIR® using an example of the iCIM manufacturing systems from Festo®. The authors specified a reduction of engineering effort by up to 50% and an up to 50% faster start of production (SOP) when all component models were available in the internal simulation model library of COSIMIR®. Additionally, a library of reusable controller programs was developed. The effort for the first system is specified to have been 30% greater, with a 20% delayed SOP. This emphasizes the importance of model libraries with standardised automation components.

This short review already shows the current options and limitations of VC for manufacturing systems. On the one hand beneficial effects such as reduced real commissioning time or improved planning quality are emphasized (Auinger et al. 1999, Zäh et al. 2006, Reinhart and Wünsch 2007, Wischnewski 2007, Rossmann et al. 2007) but on the other hand the modelling required for the virtual manufacturing system (if not just neglected) is judged by many authors to be difficult and associated with large effort (Moore et al. 2003, Park et al. 2006, Zäh et al. 2006), regardless of which simulation tool is used. The review shows moreover, a lack of accessible and 'easy to use' engineering and simulation environments which could assist the engineer to set up and conduct VC.

Thus this review indicates especially the need for improved model building methods to minimize the effort and expertise required to build a virtual manufacturing system, which can be used for checking control code and planned physical setup in a VC. This must comprise especially the set up and utilization of model libraries with standardised mechanical, electrical and control components in an engineering and simulation environment which is accessible even by SMEs.

GENERATION OF PLANT MODELS FOR VC

The investigation of possibilities for simplifying the generation and use of simulation models in a VC is a collaborative research project of the University of Applied Sciences and Arts (UASA) Hannover and the University of Glamorgan. The starting point for this project is the industrial 3D plant simulation tool CIROS®, originally developed at the institute for robot research (University of Dortmund), as a robot simulation tool COSIMIR® (Freund et al. 1994, Freund and Rossmann 1995). CIROS® allows the integrated execution of 3D mechanics with robot and control programs using either internally emulated controllers or external real or virtual controllers via OPC. Besides this, CIROS® provides features such as sensor and actuator simulation, collision detection, transport simulation for carrier based systems or AGVs, and also an XML model interface. The basic model generation concept for mechanical components within CIROS® is organized in two levels, here called high-level modelling and low-level modelling.

High-level Modelling

CIROS® allows the composition and simulation of virtual manufacturing systems based on an internal component model library containing several mechanical components including robot models. These models already contain the functional interaction of mechanical behaviour with actuators and sensors. If it is possible to compose the virtual manufacturing system from such library components – high-level modelling - it is relatively easy to set up and conduct VC, where some additional effort arises when composing the plant model within the 3D editor, configuring I/O connections and transferring controller programs.

Low-level Modelling

If there are no appropriate simulation models available in the library, the effort is far greater because it is necessary to build new models based on CAD data, which is not only a problem in CIROS® but similarly for other simulation tools such as e.g. Delmia Automation®. This low-level modelling comprises the whole functional chain, and is a non-trivial task requiring considerable modelling expertise (Park et al. 2006). Here it is necessary to carry out the geometrical, functional and electrical modelling to create a structured mechatronic component model.

Geometrical Modelling

The generation of a structured mechatronic model starts with the import of the geometry data from a CAD system. For this purpose CIROS® provides import filters for e.g. STEP and IGES. Manual simplification of overly complex geometry data may become necessary:~~.~~ The import of standard CAD data often results in an unstructured geometrical model as shown in figure 1 (①), illustrating an example component for a transportation system.

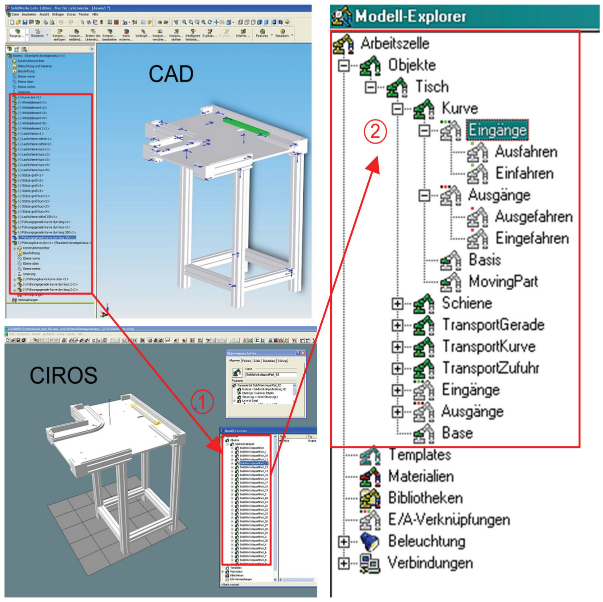


Figure 1: From CAD data to simulation object

In this case, manual hierarchical structuring of the CAD data into objects (like robots, sensors, tools or work pieces), sections (single static units e.g. joints of a robot) and components or hulls (describing the geometry e.g. cylinders, spheres or points), is the crucial step in creating the model adumbrated in figure 1- (②). If the CAD data provided are not appropriately structured e.g. with regard to moving parts, the resulting geometrical model is not directly usable for the following functional modelling, nor would be a simulation based on such model. In the worst case a CAD redesign may become necessary to provide the necessary structures in the geometry data.

Having created appropriately structured geometrical models these must be provided with functions (functional modelling) and electrical inputs and outputs (electrical modelling).

Functional Modelling

In the functional modelling stage it is necessary to manually allocate actuator functions such as translation, rotation, gripping etc. and sensor functions to selected parts of the geometrical models, which results in the definition and parameterisation of integrated functional models such as cylinder, turntable, sensor, gripper and so on.

Electrical Modelling

For the final electrical modelling it is necessary to manually add electrical inputs/outputs to the functional models for later connection to inputs/outputs of control programs thus creating complete mechatronic models.

This low-level modelling procedure can only be done by “technically experienced persons with detailed knowledge of mechanical, electrical, pneumatic, and geometric data of single components” (Wischnewski and Freund 2004).

NEW CONCEPTS FOR SIMPLIFIED VC OF MANUFACTURING SYSTEMS

The Model^{CAT} project (Hoyer et al. 2008) at the UASA Hannover demonstrated for chemical processes, that in principle it is possible to automatically generate simulation models of chemical processes for VC based on the data stored in a CAE planning tool. This allows to rapidly conduct a VC after planning, engineering and programming have been completed, at the latest, because then all necessary data for assembly and real commissioning are available in the CAE database.

This current project investigates to what extent the Model^{CAT} concept is transferable to manufacturing systems. A CAE planning tool with object oriented database will be used to hold the planning data and additional data needed for simulation (mechatronic models, control programs). The combination of various data (mechanical data, electrical data, programs if applicable, and data needed for simulation), together in a CAE planning database tool, is a novel approach. The vision is the combination of a CAE planning tool with a simulation tool like CIROS® which should ideally make it possible to assemble the mechatronic components already existing in the CAE planning tool and then to simulate the system with CIROS®. Thus, the manual high-level modelling procedure in CIROS® could be, at least, partly omitted as the plant simulation model with I/O mapping would be built automatically from the CAE planning database.

In order to validate this concept which would provide simplified generation of virtual manufacturing systems for VC, an overarching tool called the "Prototypical Engineering Modelling and Simulation Environment (PEMS)" was designed (Figure 2). CIROS® as the central simulation tool plays one key role in PEMS, the other key role the CAE planning tool COMOS® with object-oriented database (COMOS 2010). Additional off the shelf tools are used for programming PLCs, robots and HMIs.

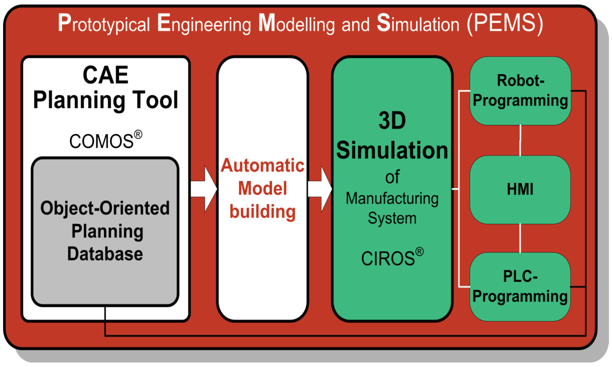

Prototypical Engineering Modelling and Simulation (PEMS)
CAE Planning Tool
COMOS®
Object-Oriented Planning Database
Automatic Model building
3D Simulation of Manufacturing System
CIROS®
Robot-Programming
HMI
PLC-Programming

Figure 2: Prototypical Engineering, Modelling and Simulation Environment

Currently, CAE tools such as COMOS® are mainly used for engineering planning and documentation purposes. The integration of data for the simulation of manufacturing systems has not been investigated hitherto.

The first task in developing PEMS is the investigation of how to automatically generate simulation models for CIROS® from the CAE database, with the aid of an additional model building tool, in the case where the component models of the virtual manufacturing system already exist in a component library (*high-level modelling*).

Another task will be the investigation of systematic set-up and extension of component model libraries, because typically projects in manufacturing systems may use new components for which simulation models do not exist in the component model library. Therefore the *low-level modelling* procedure needs systematisation where the formulation of requirements concerning function-oriented structuring of CAD data to geometrical objects by design engineers, is crucial.

In this project the set-up of exemplary simulation models in the component model library of the COMOS® CAE database will be done manually in the first step. However, in future, importing such simulation models into the component database with minimum effort, is an important strategic point: one possible future strategy is the systematic collection of models from component and subsystem vendors (Figure 3), which may allow the provision of such models together with the hardware as predicted in (Schlögl 2007).

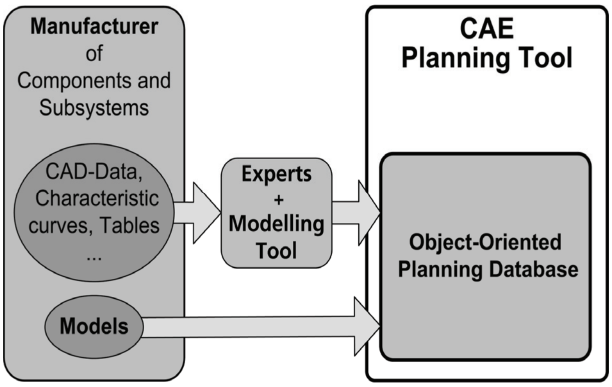

Manufacturer of Components and Subsystems
CAD-Data, Characteristic curves, Tables ...
Models
Experts + Modelling Tool
CAE Planning Tool
Object-Oriented Planning Database

Figure 3: Future component model generation strategy

A promising new approach, supporting this strategy, is the development of AutomationML® (Drath 2010) by companies like Daimler, ABB, KUKA, Rockwell Automation and Siemens along with the Universities of Karlsruhe and Magdeburg and some smaller engineering companies. AutomationML® will provide an intermediate format for automation data exchange including component model data. AutomationML® uses CAEX as top level format for the description of the topology, COLLADA for geometry and kinematics and PLCopen XML for the overall behaviour (including electrical and control). The intention is the reduction of engineering efforts and improvement of quality by interconnecting heterogeneous tools, which may become especially valuable when setting up VC with different tools and exchange of model data using AutomationML® (Drath et al. 2008a).

Test Bed Validation of the Concept

The test bed selected to investigate the "Prototypical Engineering Modelling and Simulation Environment (PEMS)" is a small pilot manufacturing cell, referred to as "Robot-based Flexible Assembly System" at the UASA Hannover. It consists of a Siemens S7 PLC controlled transportation system with autonomous track-bound transport cars, an associated elevator and two robot-based assembling units (SCARA robot and robotic palletiser) as shown in figure 4. The robotic palletiser contains a Berger-Lahr Motion controller/Soft PLC programmed with CoDeSys, the SCARA robot is programmed with a proprietary robot language. The transportation system includes an identification and data storage system which consists of data read/write stations on the tracks and mobile data carriers in the transport cars.

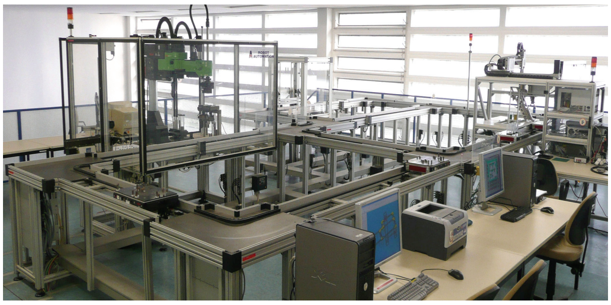


Figure 4: Robot-based Flexible Assembly System

The validation of the new PEMS approach for a simplified VC procedure will start with the low-level modelling of hardware components while following the defined modelling systematic, and storing component models of the pilot manufacturing cell in the CAE planning database of COMOS®. After re-engineering of the pilot manufacturing cell in COMOS® the next validation step will be the automated (as far as possible) model building of the virtual manufacturing cell for simulation using CIROS®. The validation will end with the execution of a VC, during which the system functions, including mechanical behaviour and control functions, will be tested by simulation. Purposely selected faults will be contrived in order to determine the efficiency of VC.

SUMMARY AND FUTURE PROSPECTS

This project is expected to demonstrate how it could become possible for control system engineers and commissioning engineers to conduct a VC for manufacturing systems based on planning data, without requiring a high level of expertise in model building and simulation. This will necessitate that CAE and simulation tools provide the appropriate functionality, especially high-level simulation model building from the CAE planning database. In addition to the improved tool environment, the object-oriented database containing simulation models of manufacturing system components, will be the subject of detailed investigations and testing. From the authors point of view it is reasonable and essential to relocate the low-level modelling effort as far as possible to the component source, i.e. to the designers, manufacturers and suppliers of components or subsystems. The manufacturers should be encouraged to provide mechatronic simulation models of their components. To support this purpose procedural methods for simplified low-level modelling strategies are investigated in the context of this project. The resulting component or subsystem models should be tool-independent, because it is not reasonable for the manufacturers to create different models for different CAE planning tools and/or simulation tools. For this purpose it will be necessary to define a standardised structure and data format for mechatronic models in future. AutomationML® is an applicable format and has the potential to support such data formats. Customer request for component simulation models, not only CAD data (which are currently often hard to use for simulation) or electronic datasheets, would accelerate this process. Experts from universities, in cooperation with component manufacturers, should make the first move to build up exemplary simulation model libraries of components or subsystems, for use in CAE environments like the proposed PEMS.

The new approach presented here, with the pilot manufacturing cell acting as test bed for the prototype implementation, will show a solution roadmap to reduce the considerable efforts required for the modelling process to such a degree that also moderately resourced small and medium-sized enterprises may become able to employ VC to optimise their engineering processes.

REFERENCES

Auinger, F., Vorderwinkler, M. and Buchtela, G. 1999. "Interface driven domain-independent modeling architecture for 'soft-commissioning' and 'reality in the loop'", *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, USA, 1, pp. 798-805.

Bani Younis, M. and Frey, G. 2003. "Formalization of existing PLC programs: A Survey", *CESA 2003*, Lille (France).

Comos Industry Solutions. 2010. COMOS, Available: http://www.comos.com [accessed 02. Feb. 2010]

Danielsson, F., Moore, P. and Eriksson, P. 2003. "Validation, off-line programming and optimisation of industrial control logic", *Mechatronics*, 13, 6, pp. 571-585.

Drath, R. (Ed.) 2010. "Datenaustausch in der Anlagenplanung mit AutomationML ", Springer Verlag, Berlin Heidelberg, pp. 326

Drath, R., Lüder, A., Peschke, J. and Hundt, L. 2008a. "AutomationML - the glue for seamless automation engineering", *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008*, Hamburg, pp. 616-623.

Drath, R., Weber, P. and Mauser, N. 2008b. "An evolutionary approach for the industrial introduction of virtual commissioning", *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008*, Hamburg, pp. 5-8.

Freund, E. and Rossmann, J. 1995. "Systems approach to robotics and automation", *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Nagoya vol.1, pp. 3-14.

Freund, E., Rossmann, J., Uthoff, J. and Can Der Valk, U. 1994. "Towards realistic simulation of robotic workcells", *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94.* , Munich, 1, pp. 39-46.

Hoyer, M., Schumann, R., Hoffmann, P. and Premier, G. C. 2008. "Virtuelle Inbetriebnahme mit Model^{CAT} - Vom Prototypen zum industriellen Einsatz / Virtual Start-Up with Model^{CAT} - From Prototypical Realisation to Industrial Implementation", *Automation 2008*, Baden-Baden, VDI-Berichte 2032, pp. 203-206.

Kühn, W. 2006. "Digital Factory - Simulation Enhancing the Product and Production Engineering Process", *Proceedings of the Winter Simulation Conference WSC 06.*, pp. 1899-1906.

Moore, P. R., Pu, J., Ng, H. C., Wong, C. B., Chong, S. K., Chen, X., Adolfsson, J., Olofsgard, P. and Lundgren, J. O. 2003. "Virtual engineering: an integrated approach to agile manufacturing machinery design and control", *Mechatronics*, 13, 10, pp. 1105-1121.

Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., Han, K. H. and Chang, M. 2006. "Development of Virtual Simulator for Visual Validation of PLC Program", *International Conference on Computational Intelligence for Modelling Control and Automation,and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06).*

Reinhart, G. and Wünsch, G. 2007. "Economic application of virtual commissioning to mechatronic production systems", *Production Engineering*, 1, 4, pp. 371-379.

RIF - Dortmunder Initiative zur rechnerintegrierten Fertigung (RIF) e. V. 2010. CIROS Engineering, Available: http://www.ciros-engineering.com/en.html [accessed 02. Feb. 2010]

Rossmann, J., Stern, O. and Wischnewski, R. 2007. "Eine Systematik mit einem darauf abgestimmten Softwarewerkzeug zur durchgängigen Virtuellen Inbetriebnahme von Fertigungsanlagen von der Planung über die Simulation zum Betrieb", *atp - Automatisierungstechnische Praxis*, 2007, 07, pp. 52-56.

Schlögl, W. 2007. "Einsatz der Digitalen Fabrik von der Anlagenplanung bis in den Laufenden Betrieb", *GMA-Kongress 2007*, Baden-Baden, VDI-Berichte 1980, pp. 717-725.

Thapa, D., Park, C. M., Dangol, S. and Wang, G.-N. 2006. "III-Phase Verification and Validation of IEC Standard Programmable Logic Controller", *International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06).*

VDI-Gesellschaft Fördertechnik Materialfluss Logistik 2008. VDI-Richtlinie 4499 - Blatt 1: Digitale Fabrik - Grundlagen

Westkämper, E., Bierschenk, S. and Kuhlmann, S. 2003. "Digitale Fabrik – nur was für die Großen?", *wt Werkstattstechnik online*, 93, 1/2, pp. 22-26.

Wischnewski, R. 2007. "Virtuelle Inbetriebnahme", *A&D Kompendium 2007/2008*, pp. 64-66.

Wischnewski, R. and Freund, E. 2004. "COSIMIR Transport: Modeling, simulation and emulation of modular carrier based transport systems", *International Conference on Robotics & Automation*, New Orleans, LA, United States, 2004, pp. 5171-5176.

Zäh, M. F. and Wünsch, G. 2005. "Schnelle Inbetriebnahme von Produktionssystemen", *wt Werkstattstechnik online*, 95, 9, pp. 699-704.

Zäh, M. F., Wünsch, G., Hensel, T. and Lindworsky, A. 2006. "Nutzen der virtuellen Inbetriebnahme: Ein experiment - Use of virtual commissioning: An experiment", *ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb*, 101, 10, pp. 595-599.

AUTHOR BIOGRAPHIES

Peter Hoffmann was born in Hannover, Germany and went to the University of Applied Sciences and Arts Hannover where he studied computer engineering and obtained his Dipl.-Ing. degree in 2000. He is now working as research staff member at the automation engineering lab at the same university, and he is currently a PhD student at the Faculty of Advanced Technology at the University of Glamorgan. His main research interest is in the field of control design for manufacturing systems and more specifically he is interested in verification and validation of PLC programs. His e-mail address is : `Peter.Hoffmann@fh-hannover.de`

Reimar Schumann received the PhD degree in Automatic Control from the Technical University Darmstadt in 1982. During his industrial career he was responsible for the design and development of a third generation digital process control system. Since 1989 he is teaching process control at the University of Applied Sciences and Arts Hannover. His current research interest is focussed on process and production control and system design. His e-mail address is : `Reimar.Schumann@fh-hannover.de`

Talal M. A. Maksoud is a Reader in the University of Glamorgan. He has over 30 years experience in research. He has published over 75 papers in Journals and international conferences. He has published several best papers in the field of advanced grinding technology. His main interests are in Grinding technology, heat transfer analyses, computational fluid dynamics, Aerodynamics, and Artificial Intelligence use in Manufacturing. Dr Maksoud has supervised over dozen PhD's as well as MPhil's. He has acted as external and internal examiner for several PhD's. He is a recognised referee for several peer reviewed academic Journals.
His e-mail address is : `tmaksoud@glam.ac.uk`

Giuliano C. Premier is a Reader in Low Carbon Systems Engineering and senior member of the Sustainable Environment Research Center (SERC). His research activities cover modelling, control, instrumentation, renewable energy systems, biological wastewater treatment. His Ph.D. concerned the modelling and control of anaerobic digestion (AD) processes and he has since been involved with research into anaerobic microbial processes. His research also includes industrial computer aided control system design (through a longstanding collaboration with the University of Applied Sciences, Hannover) and instrumentation for soil and groundwater bio-activity monitoring. He has and continues to contribute to collaborative projects with universities and industrial partners.
His e-mail address is : `gcpremier@glam.ac.uk`

RESEARCH ON SIMPLIFIED MODELLING STRATEGY FOR VIRTUAL COMMISSIONING

Peter Hoffmann [(a)], Reimar Schumann [(b)], Talal M.A. Maksoud [(c)] and Giuliano C. Premier [(d)]

[(a), (b)] University of Applied Sciences and Arts Hannover, 30459 Hannover, Germany
[(c), (d)] University of Glamorgan, Pontypridd, Wales, UK, CF37 1DL

[(a)] peter.hoffmann@fh-hannover.de, [(b)]reimar.schumann@fh-hannover.de, [(c)]tmaksoud@glam.ac.uk,
[(d)]gcpremier@glam.ac.uk

ABSTRACT

Virtual commissioning (VC) of manufacturing systems has been a research topic for more than a decade, but there are still problems in accomplishing a VC; especially in respect to the effort needed to build a simulation model. To date the design of simulation models requires a considerable effort from skilled experts in each enterprise which seeks to use VC. Small and medium-sized enterprises (SME) usually do not have those experts. This fact and the significant effort typically make VC unattractive to SMEs. To overcome this problem it is necessary to provide sufficiently detailed simulation models of manufacturing systems, in a more cost effective and accessable way. This paper proposes changes in the way simulation models are generated and presents new concepts for simplified and systematic design of manufacturing system models for VC based on standardised recipes formalising design of mechatronic component models from CAD data.

Keywords: CAD, manufacturing systems, simulation model building, virtual commissioning

1. INTRODUCTION

Nowadays in a global competition, manufacturing systems must enable cost effective and flexible production. The industrial environment is characterised by shortening of product life-cycles, increasing number of product variants and a requirement for rapid time-to-market, which leads to a progressively tightening timeframe for manufacturing system engineering and the need for better planning quality at the same time.

Manufacturing systems are composed of many off-the-shelf parts and some purpose-built parts or sub-systems, like storage, magazines, conveyors, handling and transportation systems, machining and assembling tools, robots, control and HMI systems. After the product and plant planning phase the following engineering of manufacturing systems includes the mostly sequentially executed phases: mechanical engineering, electrical engineering and automation engineering with programming of robots, PLCs and HMI (Fig. 1).

Hitherto, after completing engineering, procurement and assembly, the real commissioning is finally done, and the traditional way of testing using the real plant and the real control system is still common. Design problems and programming errors in significant quantities remain undetected before the first system start-up. As a consequence of this procedure time and money consuming corrective measures become necessary. Because of time pressure and the risk of damages only rudimental failure scenarios are tested at this stage and unidentified errors result in additional time delays and increased costs during the early production phases.

One way of overcoming this dilemma could be virtual commissioning conducted between completed engineering and assembly of production facilities. The intention of VC is to test manufacturing systems and associated control programs through simulation conducted before the real systems are realised.

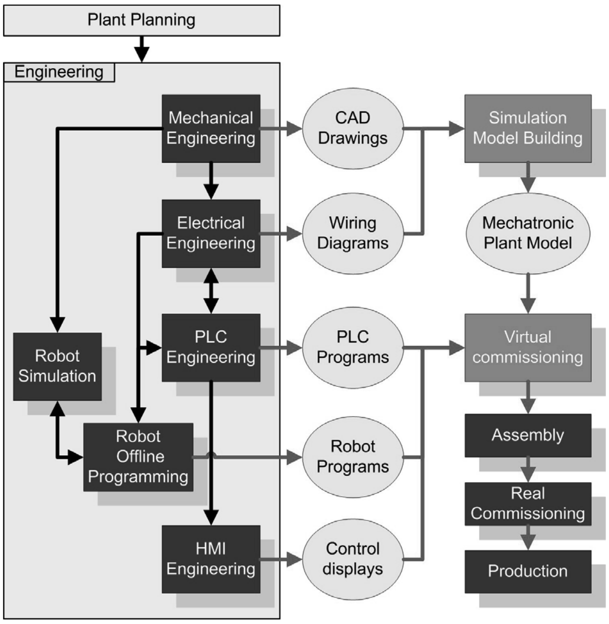


Figure 1: Engineering work-flow with VC

Conducting a VC requires a virtual manufacturing system. For this purpose a simulation model building process combining data from different engineering tools is necessary. Currently this modelling is typically done using simulation tools promoted for VC.

An important goal of VC is the early validation of PLC code in conjunction with associated mechanical and electrical design, in order to reduce the considerable time delays during commissioning (Zäh & Wünsch 2005). Delays which are caused by the error-prone control code design. This and other advantages such as reduced real commissioning time and higher quality planning are meanwhile reported by many researchers, as reviewed in e.g. (Hoffmann et al. 2010, Jain et al. 2010). Many authors report the beneficial effects of VC (Zäh et al. 2006, Reinhart & Wünsch 2007, Rossmann et al. 2007) but indicated also the great effort needed for simulation model building (Park et al. 2006, Botaschanjan et al. 2009, Zäh et al. 2006, Neugebauer & Schob 2011). The review in (Hoffmann et al. 2010) showed a need for changes in the way simulation models are generated in order to reduce the effort as well as skills needed to build a virtual manufacturing system for VC.

2. DISTINCTION BETWEEN SIMULATION AND VIRTUAL COMMISSIONING

In spite of the mechatronic configuration of modern manufacturing systems, the development process is still focused on mechanical engineering and consequentially the simulation has concentrated primarily on design and mechanical engineering; where fit and specified behaviour of mechanical plant components must be ensured.

Simulation may be conducted by starting with plant design and material flow simulation, then continuing through all engineering phases to the realisation of a virtual manufacturing system with generic or approximate and parsimonious models. Sometimes the term VC is also used for these types of simulations, but a realistic VC is not possible until detailed engineering design has been completed and the real components have been identified.

For the sake of this discussion, the authors understand VC to be the simulation of comprehensively specified manufacturing systems using virtual simulation models and the original and unmodified control programs intended for deployment on the real system – as opposed to design verification by simulation in early engineering phases.

2.1. Separate simulation of mechanics and control – How both should work

To achieve a separate verification of mechanical design, a 3D simulation which is independent of the real control programs is sufficient to test the expected and specified mechanical behaviour (i.e. with no interaction with simulated or real controller).

In separately verifying control programs, a simple simulation at I/O level is often done (without interaction with realistically simulated mechanical elements, which would reflect the specified static and dynamic behaviour of the manufacturing system e.g. with timing functions).

Such separate simulations are able to detect on one side mechanical resp. geometrical planning errors and on the other side deviations from the nominal behaviour instigated by control functions within the simulation tool.

2.2. Integrated simulation of mechanics and control – How the system will work

If the impact of control programs on the 3D mechanical behaviour of the manufacturing system is to be tested in detail and in an integrated manner, modelling and simulation of the complete functional chains is necessary. These chains would link control programs through sensors, actuators and drives onto the mechanical movements, and would include both, simulation of mechanical behaviour and of control programs. To achieve this, it is necessary to build a comprehensive mechatronic plant model (Fig.1), which should have its conceptual origin already modelled in CAD.

Such a VC is able to detect simultaneously errors in mechanical design, electrical design and control software. Typical errors have previously been identified (Neugebauer & Schob 2011). In contrast to real-world commissioning, testing of failure scenarios without endangering people or the risk of damages, is possible (Rossmann & Heinze 2010).

A VC requires a coupling between controller and the 3D plant simulation tool. This is possible through the following two configurations:

- HIL (hardware in the loop) using simulated plant and real PLC which is often realised using OPC (OLE for Process Control). The use of OPC together with simulation tools is critically analyzed in (Carlsson et al. 2012). A few tools support direct coupling via fieldbus or fieldbus emulation. Even the coupling of real robot controllers is possible within special configurations.

For the HIL configuration, the real PLC is required in advance, but VC is realisable before building the plant.

- SIL (software in the loop) using simulated plant and simulated PLCs. This is possible for different virtual robot controllers and is provided by several simulation tools. In addition to robot controllers, the plant simulation tool CIROS© (RIF 2012) provides the internal simulation of a Siemens S7 PLC or coupling to S7-PLCSIM via OPC.

The SIL configuration allows a complete VC without any hardware of the manufacturing system.

3. APPLICATION OF VC IN LARGE COMPANIES OR SME

3.1. Virtual Commissioning in large companies

In the context of the "Digital Factory" it is in principle possible to use the complex off-the-shelf factory planning suites of tools from the market-leading vendors (Delmia©, Siemens©) for a VC. These suites of tools support all phases of factory planning, but they usually require high investment costs, a high-level of training, high deployment penetration (whole departments), in-house secondment of consultants from the vendors for the implementation of custom-built functions and the laborious building of model libraries designed to the user's specifications. Therefore, only large companies (e.g. in the automotive industry) selectively choose to conduct a VC using such suites of tools. Nevertheless the simulation models of complex manufacturing systems for VC are often not available in sufficient time to be justified for this purpose.

3.2. Virtual Commissioning in SME

The possibility and the advantages of VC are generally not well known in SMEs neither are the tools for VC, apart from perhaps, as part of the "Digital Factory" solutions. Consequently, there is only limited use of these solutions by SMEs, because they generally do not have the resources to start solving their problems with such "Digital Factory" suites of tools. (Drath et al. 2008b, Westkämper et al. 2003).

The start-up costs (licences, training etc.) are very high. These suites of tools are often too complex for SME to reasonably assimilate, and the change needed, from their previous practice using simpler and independent tools to these suites of tools would not generally be plausible. There is a lack of 'easy to use' engineering and simulation environments which could assist engineers in SMEs to set up and conduct VC. The time pressure during projects prohibits the simultaneous introduction of new methods and tools and would, in conjunction with the lack of skilled experts, inhibit uptake and the prerequisite building of simulation model libraries. If the number of newly built production lines is too small, there would be no return on investment in the training of personal and the modelling efforts for one project. Also, if the next project is dissimilar to the first, the modelling would often require again high efforts, as it will not allow the reuse of the already built library models. As a result of these facts, the level of use of VC in SMEs is rare (Stern et al. 2010).

4. BUILDING SIMULATION MODELS FOR VC

Today, more than a dozen commercial simulation tools like ABB RobotStudio©, CIROS©, Delmia Automation©, InVision©, KUKASim© and VisualComponents© (3D-Create...), to name but a few, are available and applicable to VC.

In the context of this paper, the 3D plant simulation system CIROS© (RIF 2012) was chosen as an example. It was originally developed as robot simulation tool COSIMIR© at the University of Dortmund and allows HIL and SIL simulation as described in section 2.2. In addition, CIROS© provides features such as sensor and actuator simulation, collision detection, transport simulation for carrier based systems or AGVs, and also provides an XML model interface. Unfortunately, all such simulation tools require very difficult and time-consuming simulation model building.

The simulation model building procedure can be divided into two different modelling tasks:

1. (Low-level) component modelling
   If not all required component models are available in the library of the simulation tool, (which is in general the case when starting with VC) remarkable efforts become necessary to build additional models from available CAD data.
2. (High-level) plant modelling
   With CIROS© (and other comparable tools) the simulation model for the plant can be composed from component models using the simulation editor and provided that all required component models are already contained in the component model library.

By the means of high-level plant modelling it is relatively easy to set up a mechatronic plant model for VC, but even if this modelling of the virtual plant may be partially based on the plant CAD data provided by the plant designers: Especially the exact placement and the interconnection of the components is attendant on additional effort.

During the mechanical engineering of manufacturing systems CAD data of many off-the-shelf

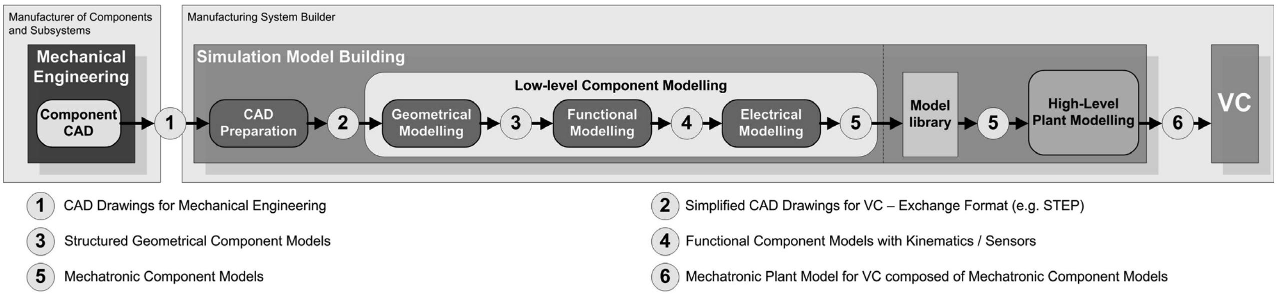

Manufacturer of Components and Subsystems
Manufacturing System Builder
Mechanical Engineering
Component CAD
Simulation Model Building
Low-level Component Modelling
CAD Preparation
Geometrical Modelling
Functional Modelling
Electrical Modelling
Model library
High-Level Plant Modelling
VC
1
2
3
4
5
6
1 CAD Drawings for Mechanical Engineering
2 Simplified CAD Drawings for VC – Exchange Format (e.g. STEP)
3 Structured Geometrical Component Models
4 Functional Component Models with Kinematics / Sensors
5 Mechatronic Component Models
6 Mechatronic Plant Model for VC composed of Mechatronic Component Models

Figure 2: Workflow from CAD Drawings to Mechatronic Plant Model

components or subsystems (supplied by manufacturers) and of some purpose-built designs (in-house work) are composed to a 3D CAD plant model.

The primary source for the component models should be the manufacturer of components and subsystems. However, nowadays the component manufacturers provide 3D CAD data of the components (at best). This means that for all components not available in the component model library of the simulation tool a CAD preparation and low-level component modelling procedure (Fig. 2) must be carried out by the user which complicates the simulation model building for VC considerably and requires specific modelling expertise.

In this paper the low-level modelling procedure of components is critically reviewed and alternative approaches are proposed. The starting point is the 3D CAD model data for the components available either from the component manufacturers or the plant designers. In both cases (plant and component modelling) the 3D CAD model data must be pre-processed and transferred from the CAD system to the simulation system to allow efficient simulation of the geometry and the identification of model structures.

4.1. CAD Data Transfer

The CAD data transfer from CAD tools to simulation tools has other requirements than the data exchange between different CAD/CAM tools used in manufacturing engineering. Not necessary for VC are e.g. detail data like finish specification or tolerances, material data in contrast may make sense, e.g. for the simulation of an inductive sensor.

A big problem exists to date at the interface between the different CAD tools and simulation tools for VC (Fig. 2, ②). Even if functionally structured CAD models are available, the data interfaces allow in general only partial transfer (without kinematics) of this structure information.

In order to transfer kinematics additionally to geometry it is possible to use API (Application Programming Interface) functions for access to internal data of the CAD tool. Such a solution was demonstrated by (Neugebauer & Schob 2011). The disadvantage of this approach is the necessary programming and software maintenance of each combination of CAD tool and simulation tool.

From the authors point of view it is more promising to further develop suitable CAD exchange formats. There are a lot of established exchange formats with special advantages available (Friedewald et al. 2011, Fröhlich 2011), but none has established for exchange of kinematics.

In principle suitable formats for the exchange of geometry and kinematics are STEP (**ST**andard for the **E**xchange of **P**roduct model data) and the Automation Markup Language - AutomationML (Drath et al. 2008a).

STEP supports kinematics since 1996 with its application protocol AP214, Part105 (Haenisch et al. 1996), but to date there is no industrial implementation in a tool. First approaches are presented by (Hedlind et al. 2011, Li et al. 2011).

AutomationML is a neutral, intermediate data format based on XML for automation data storage and exchange including component model data, not limited to geometry and kinematics.

The intention of AutomationML is the reduction of engineering efforts and quality improvement by interconnecting heterogeneous tools, which may become especially valuable when setting up VC with different tools and exchange of model data using AutomationML.

AutomationML combines several already existing standards respectively data formats.

- CAEX (IEC 62424) is used as top level format for the description of the topology and hierarchical structure of objects used in the manufacturing system (including necessary properties and relations between objects).
- COLLADA for describing the 3D geometry with mechanical interconnections and dependencies. The possibility of exchanging kinematics between different tools (e.g. 3D CAD and simulation tools for VC) is an important advantage compared to the exchange formats like STEP used today, where only exchange of geometry is implemented in tools.
- PLCopen XML is used for the description of overall behaviour (including electrical and control information like I/O relations). Regarding the term “behaviour” it has to be distinguished between the internal behaviour of physical objects and the description of PLC code for controlling physical objects.

A first implementation of COLLADA import with kinematics into a simulation tool (ABB RobotStudio©) is presented in (Kuhlenkötter et al. 2010).

4.2. CAD Preparation

3D CAD data are the basis for the building of the simulation models, and CAD preparation is the first step for the design of mechatronic component models and thus important for VC. The 3D CAD data delivered from the manufacturers of components and subsystems or generated during plant design are not in general directly suitable for use in a VC simulation because most CAD models are geometrically too complex. This is why, a thorough model analysis and simplification of the CAD model data is usually necessary especially to reduce calculations. The goal is to reduce the number of details (geometrical elements) resulting in a reduced number of facets in the simulation model.

For example, the CAD data of aluminium profiles, used for many constructions within manufacturing systems (Fig. 6), must definitely not be used for simulation without simplification. Their complex structures (Fig. 3) could extremely increase number of facets and extend calculation time.

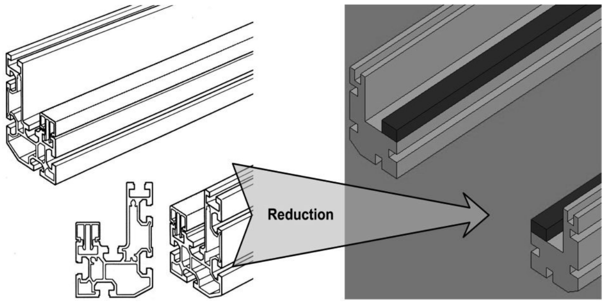


Figure 3: Example for simplified CAD from Bosch CTS© transport system

The complexity of CAD data is not the sole problem; the inner structure of a CAD models has often to be changed as well (to e.g. reduce calculation loading in the simulation and for better model handling inside simulation tools).

As consequence, the preparation of 3D CAD data for simulation has two requirements:

1. The CAD data complexity must be reduced to make real-time 3D rendering and visualisation possible. Even if the simulation tool is able to calculate independently the 3D visualisation, the visualisation should not differ that much. Deviation would hinder the human visual analysis of the operations in the virtual manufacturing system. This ability is simulation tool dependent; e.g. CIROS© provides configurable visualisation rates.
2. The model structure should consider actuating elements and sensors. The dependencies of objects moving together/separately or being stationary, and which geometrical objects are sensors, are all relevant and important.

A small example is shown in Fig. 4. The model structure has to consider a moving object (stopper) and two sensors.

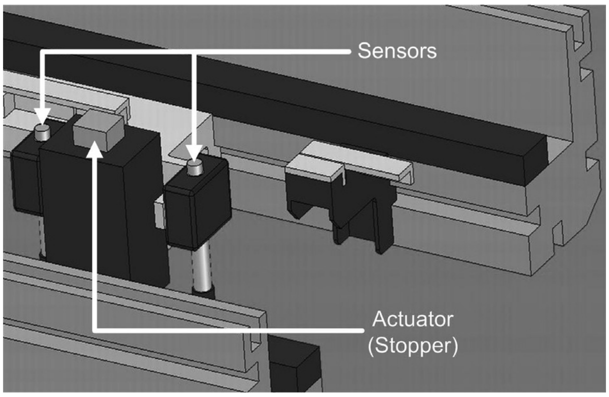


Figure 4: Stopper unit with sensors from Bosch CTS© transport system

If the CAD data provided are not appropriately structured especially with separation of moving parts, the data is not directly usable for the following low-level component modelling, nor is it possible to base a simulation on such model. In worst case, a CAD redesign may become necessary to provide in the geometry data the inner structures required for simulation.

In principle the preparation of CAD data can be done inside the simulation tool (see section 5.1, Fig. 5), which is often the case today, but then dispensable data has to be deleted and the model has to be restructured there with cumbersome detail work.

More efficient and overall faster would it be to prepare the 3D CAD data already inside CAD or simplification respectively conversion tools. In comparison with these tools the relevant simplification capabilities of simulation tools are limited and less controllable and the restructuring work is affiliated with more effort.

Modern CAD tools already provide several automated simplification features with adjustable filters to remove irrelevant geometrical features and irrelevant geometrical objects. If this functionality is not sufficient, it is possible to use specialised simplification respectively conversion tools such as: CADdoctor© (Elysium 2012), CADfix© (ITI-Transcendata 2012) or PolyTrans©/Nugraf© (Okino 2012).

Regarding the requirements mentioned above, the workflow for CAD model preparation for VC should comprise the following steps.

4.3. Simplification

Simplification can be done by removing features and removing parts.

- Feature Suppression: Removal of geometrical features irrelevant for simulation such as:
  - Holes, bosses, pockets, breakthroughs (not round), labels
  - Fillets, chamfers, roundings
  - Ribs, steps, slots
- Object Filtering: Removal of irrelevant geometrical objects like such as:
  - Hidden invisible parts
  - Small or other selected parts

The waiving of elaborate textures, respectively the definition as separate objects which can be deleted easily, is reasonable to allow further shortening of calculation time.

4.4. Component Structuring

Adjusting the CAD model inner structure is the crucial factor for CAD preparation. The model has necessarily to be separated to:

- static units
- moving units
- sensors

Up to now the building of assemblies in mechanical engineering is not done considering the functional interaction of parts, but rather aspects like common manufacturing are respected. Sometimes, that will lead to assemblies wherein static parts are mixed with moving parts. This was reported e.g. in (Hollander & Sappei 2011).

Assemblies created in mechanical engineering are often static, or all parts of an assembly move together. Converting such assemblies to single parts makes sense to minimise the calculation needs of simulation.

While working with CIROS© this practise has also proven advantageous for better handling of models during functional component modelling, because much less objects appear in the model editor.

Sensors have to be separate parts, regardless of belonging to a static unit or a moving unit.

The result of this preparation procedure will be a simplified component CAD model structured according to mechatronic considerations suitable for low-level component modelling.

After revising the CAD model, in respect to the feature objects and the structure to be adopted, conversion from exact model in a meshed surface model must then be done.

4.5. Tessellation/Meshing

The original CAD model is tessellated into a mesh of polygons, often done by triangulation to build a triangular mesh. Generally, it is essential to reduce the quantity of polygons for simulation purposes. Mesh simplification is a separate area of research for a long time and extensive studying originated a lot of techniques and algorithm. Basically, they can be divided into two groups (Kwak et al. 2010):

- Iterative coarsening of the complete mesh by removing polygons until a specified goal (e.g. number of polygons) is achieved
- Iterative refinement of a newly generated mesh (based on an initial approximation) by inserting polygons

The research in (Kuhlenkötter et al. 2010) shows the influence of quantity of facets and type of mesh (polygonal/triangular) on calculation time using ABB RobotStudio© as example. Unsurprisingly, the import duration and calculation time for collision detection ascends with quantity of facets, but the conducted experiments with a COLLADA importer offer big differences between polygonal meshes and triangular meshes. Ascending polygon quantity shows steeply rising import duration shortly exceeding one hour and leading to problems of insufficient RAM. In contrast, importing triangular meshes shows a gently inclined curve. Simulation tools for VC make use of collision detection between geometrical objects, and here as well triangular meshes show the advantage of significant shorter calculation time.

These findings allow the assumption to preferably use triangular meshed models for VC, and to limit the quantity of triangles as much as reasonable. Especially geometrical objects to be checked against each other with collision detection should contain as few triangles as possible.

CAD data providing different LOD (levels of detail) would be beneficial for simulation purpose. CIROS© for example supports different, changeable LOD.

5. RECIPES FOR LOW-LEVEL MODELLING

5.1. Conventional Low-Level Component Modelling from CAD Data (at the example CIROS©)

The starting point for this procedure is the 3D CAD model data for the components available either from the component manufacturers or the plant designers.

Building a low-level mechatronic component model in the conventional way means to describe the whole functional chain of the model, and can be divided into three stages: geometrical, functional and electrical modelling. The carrying out of this non-trivial procedure requires considerable modelling expertise and effort. Especially SMEs usually do not have the needed modelling experts.

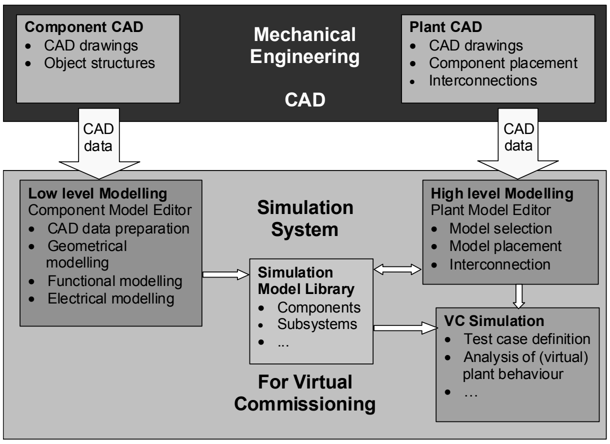


Figure 5: Simulation model generation: conventional recipe

5.1.1. Geometrical Modelling

Building mechatronic simulation models starts with the import and preparation of CAD drawings. For this purpose CIROS© currently provides import filters for STEP (AP203/214), STL, VRML and IGES. Simplification of exceedingly complex geometric data and restructuring would become necessary, if this had not been done in an external CAD preparation. CIROS© provides merging, aligning and optimisation of CAD data.

Adjusting the inner structure of these models considering static parts, actuating elements and sensors is the crucial step in creating the model. As mentioned above, the import of standard CAD data supplied from hardware vendors or in-house mechanical engineering often results in an unstructured or unsuitably structured geometrical model. Having created appropriately structured geometrical models, these must be equipped with functions (functional modelling) and electrical inputs and outputs (electrical modelling).

5.1.2. Functional Modelling

As indicated in section 4.1 today it is not or only partially possible to transfer the inner structure of the 3D CAD model from the CAD to the simulation system, so it is necessary to manually equip the geometrical models with kinematics and when indicated with sensor functions.

CIROS© provides for this purpose several actuator functions such as translation, rotation, gripping and different types of sensor functions e.g. ultrasonic, optical, capacitive, inductive or light barrier. It is necessary to manually allocate actuator and sensor functions to the respective parts of the geometrical models. This results in the definition of integrated functional models such as push cylinder, rotational cylinder, turntable, sensor, gripper and so on. These models need parameterisation e.g. stroke and speed of cylinders or timing, measurement range, switch distance and hysteresis of sensors.

5.1.3. Electrical Modelling

For the final electrical modelling, it is necessary to manually assign electrical inputs and outputs to sensors and actuators in the functional models. CIROS© allows interactive, graphical editing of these connections. Later these I/Os will be linked to I/Os of control programs thus creating complete mechatronic simulation models.

5.2. Alternative Low-Level Component Modelling

To reduce the modelling effort for the user the following alternative approaches may be taken.

5.2.1. Component simulation model to be provided by manufacturer

For electrical CAE systems the component simulation models are generally provided by the component manufacturers. So in the long term a similar approach should be taken also for the components of manufacturing systems, i.e. the simulation models should be provided by the component manufacturers. This means that the manufacturers (and not the users) will have to handle the low-level modelling of their components. The motivation to undertake this effort can only be provided by the users, especially by the big users of simulation tools which may define the provision of simulation models as a general delivery condition in their commercial terms.

5.2.2. Low-level modelling during CAD design

The component designers have a clear view of the functional structure of the component. Today, however, only part of this general functional view is implemented in CAD models which represent predominantly CAD drawings perhaps with some limited object specifications. This means that the additional functional knowledge of the designer must be documented separately or transferred from the designer to later users individually.

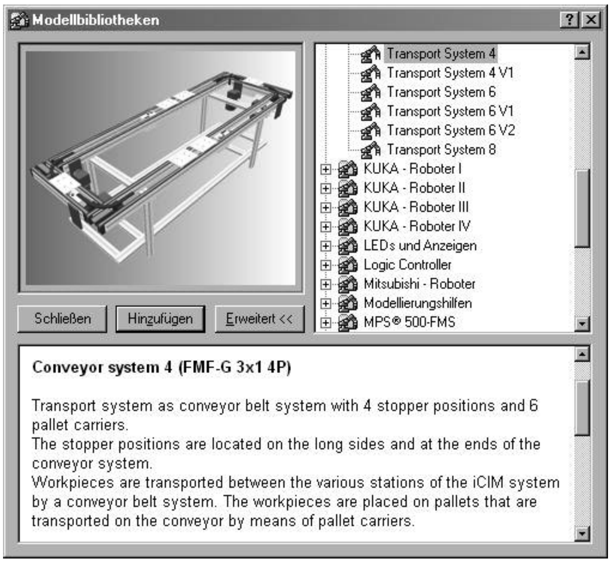
This is why the authors propose that:

- In future versions of CAD systems the functional and structure information of components must become an integral part of CAD design.
- The education of future designers should consider the structural organisation and inclusion of functional information (as required among others for the simulation models) in the CAD data. This should result in CAD model data with sufficient functional and structure to feed also simulation systems directly.
- In addition the CAD data interfaces must be empowered to transfer not only geometrical but also functional information (kinematics).

6. RECIPES FOR HIGH-LEVEL PLANT MODELLING

6.1. Conventional Recipe for High-Level Plant Modelling

The engineering of manufacturing plants is in general documented using CAD systems showing the placement of components in 3D and the interconnections of them. Simulation systems like CIROS© allow the aggregation and simulation of virtual manufacturing systems based on components in their internal library containing several mechatronic components or subsystems (for CIROS© e.g. from Festo© FMS such as conveyor systems (Fig. 6), assembly stations, handling stations and stocks, even robot models from different vendors (ABB©, FANUC©, KUKA©, Mitsubishi© etc.) have already been included. A similarly concept uses e.g. ABB RobotStudio© with so called "Smart Components", but regarding robots it is limited to the use of ABB© products.



Conveyor system 4 (FMF-G 3x1 4P)

Transport system as conveyor belt system with 4 stopper positions and 6 pallet carriers.
The stopper positions are located on the long sides and at the ends of the conveyor system.
Workpieces are transported between the various stations of the iCIM system by a conveyor belt system. The workpieces are placed on pallets that are transported on the conveyor by means of pallet carriers.

Figure 6: Example subsystem from CIROS© model library

These library models already contain the functional interaction of mechanical behaviours for actuators and sensors. If it is possible to compose the mechatronic plant model from such mechatronic components – high-level plant modelling - it is relatively easy to set up and conduct VC, because it will only require little additional effort when composing the plant model within the 3D editor and configuring I/O connections.

However, nowadays the geometrical information for the placement of the components and for their connections must be manually transferred from the CAD drawings provided by plant engineering.

6.2. Improved Recipe for High-Level Plant Modelling

In order to simplify high-level plant modelling for simulation the placement and interconnection information should be collected comprehensively already during CAD and transferred automatically to the simulation system such that the placement and interconnection of the components in the simulation model is done automatically (Fig. 7). This can be accomplished either by using improved data interfaces or by integration of CAD and simulation tool.

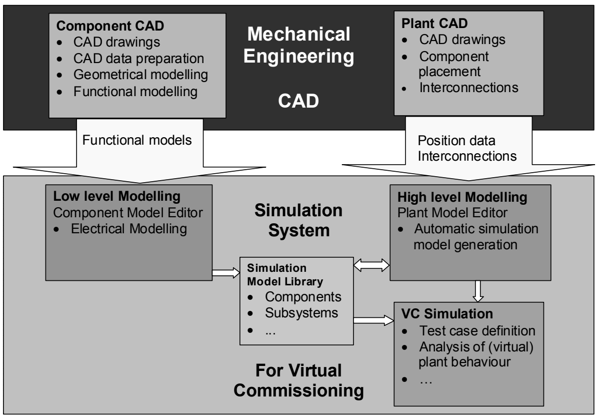


Figure 7: Simulation model generation: alternative recipe

The proposed approach of providing mechatronic models by the manufacturers of components or subsystems is supported by AutomationML.

7. SUMMARY AND FUTURE PROSPECTS

Currently, the simulation model building is carried out manually by the users of VC. It is not entirely possible until the engineering of manufacturing system is completely finished and all components are accurately specified. The necessary and laborious preparation of CAD models and low-level component modelling tends to result in serious time pressures. This may reach the point where delayed completion of the mechatronic plant models might result in their delivery after the real-world manufacturing system has been built.

An important first step towards simulation model building to be a task for control system engineers and commissioning engineers from SMEs would be the unimpeded transfer of CAD data to component modelling. The data flow should facilitate and be directly applicable to low-level component modelling suitable for VC. Therefore, it is required that designers in CAD departments in components and subsystems manufacturing enterprises and designers from enterprises engineering, building and/or commissioning manufacturing systems consider the supply of such data. The supply of appropriate CAD data would greatly reduce the laborious CAD preparation always repeated at every use to a one-time activity. The low-level component modelling which remains necessary nowadays would be better facilitated and simplified. To achieve this goal there is a need for information to be gathered from designers creating CAD models, persuasion the manufacturers of components and subsystems to provide additional CAD data is recommended to facilitate component models for VC. In order to advance thinking in mechatronic units, CAD education in universities should address model transfer, reduction and reuse for low-level component modelling.

The future goal should be the availability of complete mechatronic models (like the models available today at end of low-level component modelling) provided by the manufacturers of components and subsystems. Already today the manufacturers should change over to provide 3D CAD data suitable for VC; this would be a good basis for future building of mechatronic models. Besides an appropriate exchange format like AutomationML, joint efforts for standardisation (e.g. model content) will be necessary for this purpose.

REFERENCES

Botaschanjan, J., Hummel, B., Hensel, T. and Lindworsky, A., 2009. Integrated behavior models for factory automation systems, *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, pp. 1-8.

Carlsson, H., Svensson, B., Danielsson, F. and Lennartson, B., 2012. Methods for Reliable Simulation-Based PLC Code Verification, *Industrial Informatics, IEEE Transactions on*, 8, 2, pp. 267-278.

Drath, R., Lüder, A., Peschke, J. and Hundt, L., 2008a. AutomationML - the glue for seamless automation engineering, *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008*, 15-18 Sept. 2008, Hamburg, pp. 616-623.

Drath, R., Weber, P. and Mauser, N., 2008b. An evolutionary approach for the industrial introduction of virtual commissioning, *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, Hamburg, pp. 5-8.

Elysium, 2012. *Interactive geometry verification and healing for multi-CAD data exchange, geometry simplification for CAE, plus tools for*

Rapid Prototyping and Reverse Engineering, Available from: http://www.elysiuminc.com/Products/caddoctor.asp [accessed July 10th, 2012]

Friedewald, A., Lödding, H., Lukas, U. F. V., Mesing, B., Roth, M., Schleusener, S. and Titov, F., 2011. *Benchmark neutraler Formate für den prozessübergreifenden Datenaustausch im Schiffbau*, Fraunhofer IGD.

Fröhlich, A., 2011. *3D-Formate im Engineering-Umfeld – ein Vergleich*, Whitepaper, Darmstadt, PROSTEP AG.

Haenisch, J., Kroszynski, U., Ludwig, A. and Sørensen, T., 1996. *Specification of a STEP Based Reference Model for Exchange of Robotics Models : Geometry, Kinematics, Dynamics, Control, and Robotics Specific Data*.

Hedlind, M., Klein, L., Li, Y. and Kjellberg, T., 2011. Kinematic structure representation of products and manufacturing resources, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Athens, Greece.

Hoffmann, P., Schumann, R., Maksoud, T. M. A. and Premier, G. C., 2010. Virtual Commissioning of Manufacturing Systems - A Review and new Approaches for Simplification, *Proceedings of the 24th European Conference on Modelling and Simulation*, June 1st - 4th, 2010, Kuala Lumpur, Malaysia, pp. 175-181.

Hollander, A. and Sappei, S., 2011. *Virtual preparation of Tetra Pak Filling Machine,* Göteborg, Sweden, CHALMERS UNIVERSITY OF TECHNOLOGY, Master Thesis

Iti-Transcendata, 2012. *Solutions for CAD/CAM/CAE/PLM interoperability*, Available from: http://www.transcendata.com/products/cadfix/index.htm [accessed July 11th, 2012]

Jain, A., Vera, D. A. and Harrison, R., 2010. Virtual Commissioning of Modular Automation Systems, *10th IFAC Workshop on Intelligent Manufacturing Systems (2010)*, July 1st - 2nd, 2010, Lisbon, Portugal.

Kuhlenkötter, B., Schyja, A., Hypki, A. and Miegel, V., 2010. Robot Workcell Simulation with AutomationML Support - An Element of the CAx-Tool Chain in Industrial Automation, *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 7-9 June 2010, Munich, Germany, pp. 1076-1082.

Kwak, J.-G., Park, S. and Chang, M., 2010. Geometric data simplification for a virtual factory, *The International Journal of Advanced Manufacturing Technology*, 50, 1, pp. 409-418.

Li, Y., Hedlind, M. and Kjellberg, T., 2011. Implementation of kinematic mechanism data exchange based on STEP, *Proceedings of the 7th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Athens, Greece, pp. 152-159.

Neugebauer, R. and Schob, U., 2011. Reducing the model generation effort for the virtual commissioning of control programs, *Production Engineering*, 5, 5, pp. 539-547.

Okino, 2012. *Okino PolyTrans and NuGraf, Professional 3D Translation/Translator and Conversion/Converter Software for CAD, NURBS, Skinning, DCC and Animation*, Available from: http://www.okino.com/default.htm [accessed February 29th, 2012]

Park, C. M., Bajimaya, S. M., Park, S. C., Wang, G. N., Kwak, J. G., Han, K. H. and Chang, M., 2006. Development of Virtual Simulator for Visual Validation of PLC Program, *International Conference on Computational Intelligence for Modelling Control and Automation,and International Conference on Intelligent Agents,Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*.

Reinhart, G. and Wünsch, G., 2007. Economic application of virtual commissioning to mechatronic production systems, *Production Engineering*, 1, 4, pp. 371-379.

Rif, 2012. *CIROS Engineering*, Available from: http://www.ciros-engineering.com/en/home/ [accessed May 11th, 2012]

Rossmann, J. and Heinze, F., 2010. Modeling and simulation of malfunctions in automation systems, *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, 13-16 Sept. 2010, pp. 1-8.

Rossmann, J., Stern, O. and Wischnewski, R., 2007. Eine Systematik mit einem darauf abgestimmten Softwarewerkzeug zur durchgängigen Virtuellen Inbetriebnahme von Fertigungsanlagen von der Planung über die Simulation zum Betrieb, *GMA-Kongress 2007*, 12th and 13th June 2007, Baden-Baden, VDI-Berichte 1980, pp. 707-716.

Stern, O., Hoffmann, P. and Schumann, R., 2010. KMU und VIBN in der Fertigungstechnik, *Meeting of VDI/VDE-GMA FA 6.11 CACE*, Dec. 1st, 2010, Ladenburg.

Westkämper, E., Bierschenk, S. and Kuhlmann, S., 2003. Digitale Fabrik – nur was für die Großen?, *wt Werkstattstechnik online*, 93, 1/2, pp. 22-26.

Zäh, M. F. and Wünsch, G., 2005. Schnelle Inbetriebnahme von Produktionssystemen, *wt Werkstattstechnik online*, 95, 9, pp. 699-704.

Zäh, M. F., Wünsch, G., Hensel, T. and Lindworsky, A., 2006. Nutzen der virtuellen Inbetriebnahme: Ein experiment - Use of virtual commissioning: An experiment, *ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb*, 101, 10, pp. 595-599.

AUTHORS BIOGRAPHY

Peter Hoffmann studied computer engineering at the University of Applied Sciences and Arts Hannover and obtained his Dipl.-Ing. degree in 2000 and is now working as research staff member at the automation engineering lab at the same university. In addition he is currently a PhD student at the Faculty of Advanced Technology at the University of Glamorgan. His main research interest is in the field of control design for manufacturing systems and more specifically he is interested in verification and validation of PLC programs

Reimar Schumann received the PhD degree in Automatic Control from the Technical University Darmstadt in 1982. During his industrial career he was responsible for the design and development of a third generation digital process control system. Since 1989 he is teaching process control at the University of Applied Sciences and Arts Hannover. His current research interest is focussed on process and production control and system design.

Talal M. A. Maksoud is a Reader in the University of Glamorgan. He has over 30 years experience in research. He has published over 75 papers in Journals and international conferences. He has published several best papers in the field of advanced grinding technology. His main interests are in Grinding technology, heat transfer analyses, computational fluid dynamics, Aerodynamics, and Artificial Intelligence use in Manufacturing. Dr Maksoud has supervised over dozen PhD’s as well as MPhil’s. He has acted as external and internal examiner for several PhD’s. He is a recognised referee for several peer reviewed academic Journals.

Giuliano C. Premier is a Professor in the University of Glamorgan and senior member of the Sustainable Environment Research Center (SERC). His research activities cover modelling, control, instrumentation, renewable energy systems, biological wastewater treatment. His Ph.D. concerned the modelling and control of anaerobic digestion (AD) processes. His research also includes industrial computer aided control system design (through a longstanding collaboration with the University of Applied Sciences, Hannover). He has and continues to contribute to collaborative projects with universities and industrial partners.