INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15-18, 2005

APPLYING THE NEW VDI DESIGN GUIDELINE 2206 ON MECHATRONIC SYSTEMS CONTROLLED BY A PLC

Jens Bathelt, Anders Jönsson, Christian Bacs, Stefan Dierssen and Markus Meier

Abstract

When developing mechatronic systems it is important to consider the control, electrical and mechanical aspects throughout the development process. The new VDI guideline 2206 [1] provides a useful frame for the design of any kind of mechatronic system. This work describes a customization of that guideline when focusing on the development of systems controlled by a Programmable Logic Controller (PLC).

Predefined process modules are defined, relevant methods and tools are identified, and the interdisciplinary data flow is described. The relevance of the methods and tools used in the work is demonstrated by an industrial case study.

The case study shows the possibilities for an improved product development process using the enhanced methodology presented in this paper. However, the data flow needs to be further detailed and automated to be more efficient. The enhanced methodology developed in this work is a promising base for an efficient development process for mechatronic systems controlled by a PLC.

Keywords: Concurrent interdisciplinary engineering, Guideline, Mechatronics, Programmable Logic Controller (PLC), VDI 2206.

1. Introduction

The virtual machine concept described by the authors in [10], [11] and [9] is one reason for the adaptation of the VDI guideline 2206. This virtual machine concept consists of the three bidirectional connected components “real control”, “machine simulation” and “3D visualization” as shown in 2. The virtual machine enables a virtual initial operation of the machine.

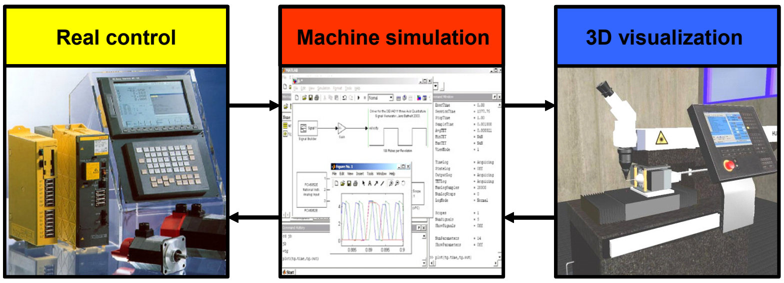


Figure 1. Concept of the virtual machine

The authors experience gained from the virtual machine led to research on interdisciplinary development of mechatronic systems controlled by a PLC ([2], [6]). The major engineering domains involved in the development process are control and mechanics. The control engineer develops the software for the PLC and the mechanical engineer designs the machine in a CAD system.

The aim is to reach – similar to the V model (Figure 3) from [1] – a common concept phase and a synchronized embodiment design as shown in Figure 2.

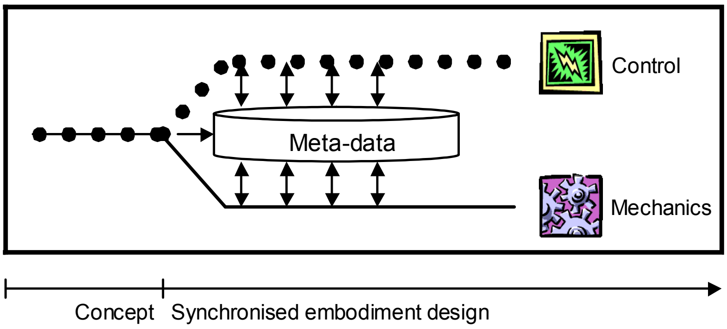


Figure 2. Motivation for the work presented in [2] and [6]

An interdisciplinary description language for the concept phase is elaborated. The transition between the design phases is described to establish a concurrent interdisciplinary design. The adapted guideline in the following section is also based on these results developed together with industry.

The new VDI guideline 2206 [1] provides a useful frame for the design of any kind of mechatronic system. It consists essentially of three elements:

- A general problem-solving cycle on the micro-level.
- The V model on the macro-level (Figure 3).
- Predefined process modules for handling recurrent working steps in the development of mechatronic systems.

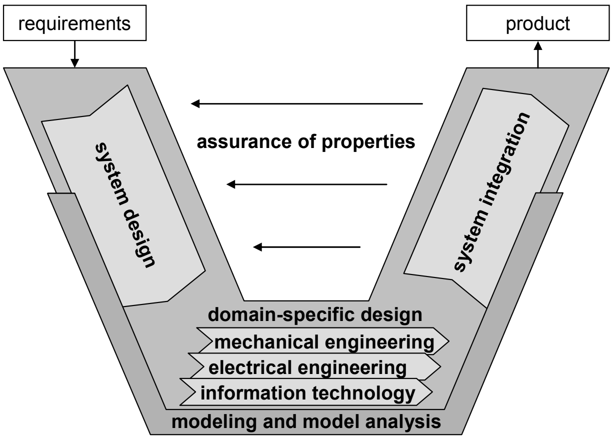


Figure 3. The V model in the VDI guideline 2206

The presented problem-solving cycle in [1] is intended to support the product developer to work on predictable subtasks as well as unforeseeable problems. The V model in [1] is adopted from software development and adapted to the requirements of mechatronics. This work is presenting a more specific adaptation on mechatronic systems controlled by a PLC in section 2. Process modules for the recurrent working steps "system design", "modeling and model analysis", "domain-specific design", "system integration" and "assurance of properties" observable in Figure 3 are described in [1]. The existing process modules are extended and adapted in the next chapter according to the specifics of the discussed product. In addition two process modules are introduced to describe the transition from and to the domain-specific design. This work aims to customize the guideline for systems controlled by a PLC like textile or packaging machines. The specific properties of those products are exploited when describing the design phases and the transitions in-between the design phases.

2. Adapted Guideline

The adapted V model is shown in Figure 4. Since the engineering domains control (developing the software for the PLC) and mechanics (using CAD to design the geometry) has been identified as the major engineering domains [2]/[6] for the discussed product, the third domain "electrical engineering" is neglected in this work. The resulting product is in addition specified in the upper right corner of Figure 4.

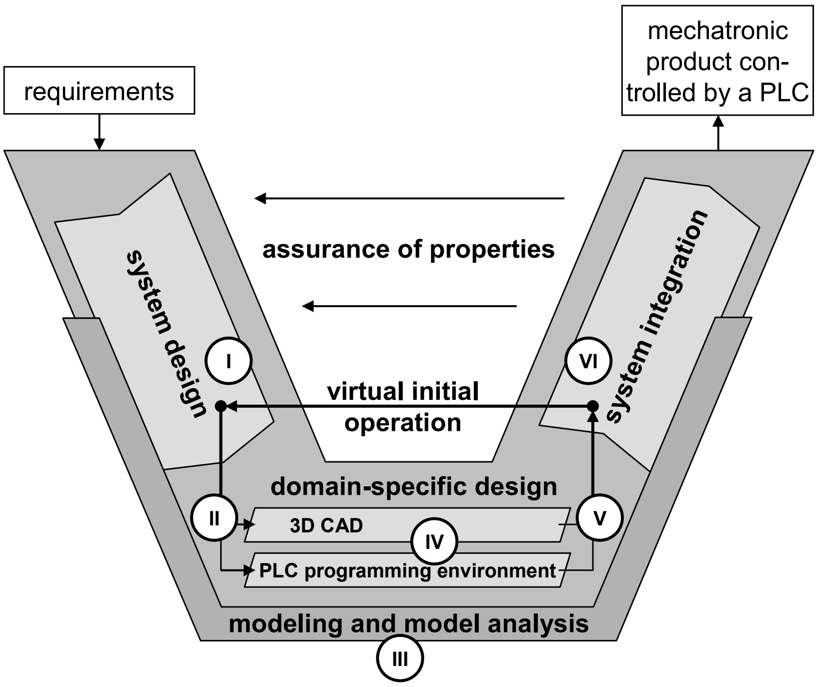

requirements
mechatronic product controlled by a PLC
assurance of properties
system design
system integration
I
VI
virtual initial operation
domain-specific design
II
3D CAD
IV
V
PLC programming environment
modeling and model analysis
III

Figure 4. The adapted V model

The adaptation of the predefined process modules for recurrent working steps in I, III, IV and VI (Figure 4) is described in the following sections. Two new process modules for the working steps in II and V are introduced to specify the transition in-between the design phases for the discussed type of mechatronic system.

The following sections will only describe the difference between the VDI guideline 2206 and the proposed changes/extensions. The intention is not to replace sections from the guideline but to complement it. This implies in particular that all statements in the guideline concerning the domain "electrical engineering" and "macro cycles" remain unchanged.

2.1 Interdisciplinary description language (I) in the system design

The predefined process module for the system design describes the conceptual design phase according to [3] and [5]. The result of this predefined process module is the interdisciplinary concept. A slight change is proposed in this work: an extension of the function structure, which leads to the process module shown in Figure 5.

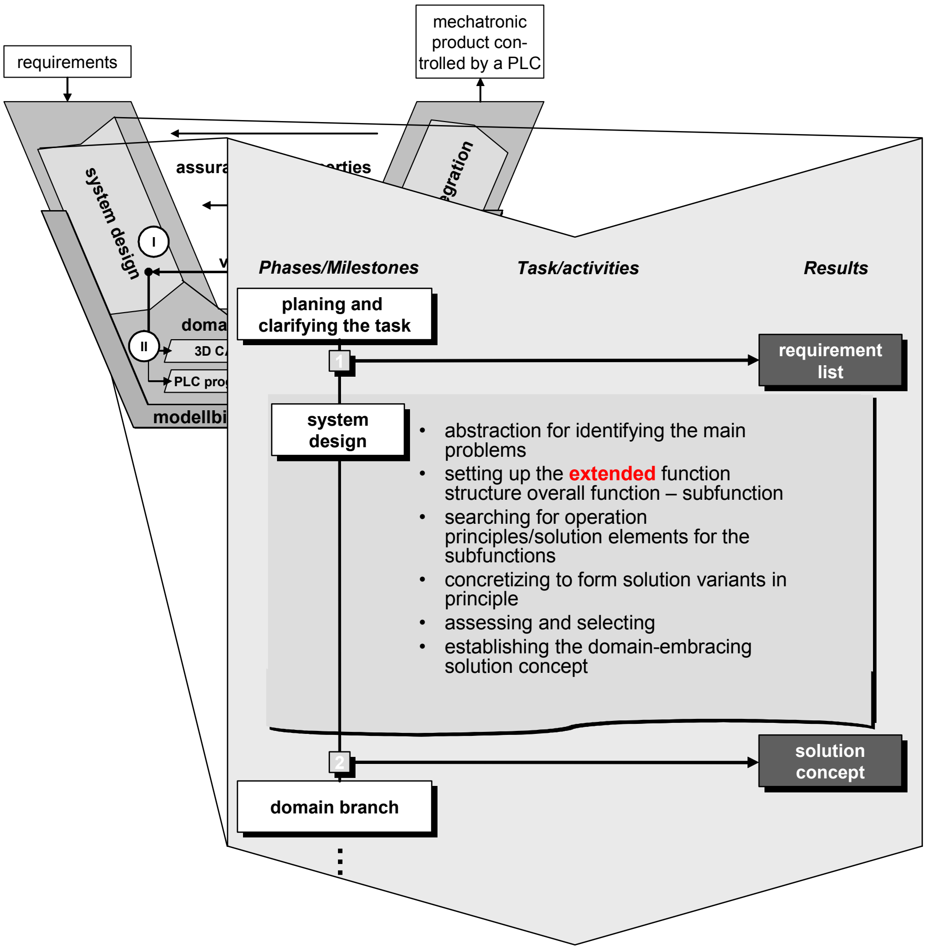

requirements
mechatronic product controlled by a PLC
assura
erties
system design
I
II
doma
3D C
PLC pro
modellbi
gration
Phases/Milestones
Task/activities
Results
planing and clarifying the task
requirement list
system design
abstraction for identifying the main problems
setting up the extended function structure overall function – subfunction
searching for operation principles/solution elements for the subfunctions
concretizing to form solution variants in principle
assessing and selecting
establishing the domain-embracing solution concept
solution concept
domain branch

Figure 5. Predefined process module for the conceptual design phase using the extended function structure

To reflect the sequential behavior and the transitions of mechatronic systems controlled by a PLC, transition conditions are introduced in the extended function structure (EFS). They indicate that a specific condition must be fulfilled before the next function can be carried out. The EFS as presented in [2] does prepare the search for solution elements as claimed in Figure 5. Beyond the traditional function structure with hierarchy and the flow variables material, energy and information [3]/[1], the EFS indicates in particular where actuators and sensors have to be used and prepares the selections of those mechatronic elements on the border between software and mechanics.

Another change in the picture is the new task called domain branch to initiate the domain-specific design. This branching is the topic of the next section. The EFS is the base for the newly developed interdisciplinary description language consisting of:

- EFS
- I/O list (listing the **i**nput and **o**utput variables of the PLC and their properties)
- S/A list (listing all **s**ensors and **a**ctors of the mechatronic system and their properties)
- Path/Step Diagram
- Sketches
- M/R data (declaring a valid range of the **m**achine parameters and defining the **r**ecipes, consisting of a set of values specifying the machine parameters)
- Mechanical main modules

The I/O list consists of all input and output variables used in the software to access the sensors and actors stored in the S/A list. The Path/Step Diagram is optional and illustrates the correlation between the actors and sensors for each step in a sub system. Many target values from the requirement list (Figure 5) can be collected in the machine and recipe data (M/R data). The machine data specifies a valid range of values to operate the machine. They help to select solutions for the functions in the function structure, in this case actors and sensors. Valid values are chosen for each recipe. The recipe data will be used in the software to run the machine in all needed different modes. The mechanical main modules are initiating the modularization in mechanics and the assembly tree for the CAD system. The three heuristic rules “dominant flow”, “branching flow” and “conversion-transmission” [19] can be used to derive potential mechanical modules from the EFS.

All entities in the interdisciplinary description language are profoundly linked. The MS Office tools Visio for the 2D-graphics and Excel as a DB are used in the case study presented in chapter 3 to store the digital data of the concept as shown in Figure 6.

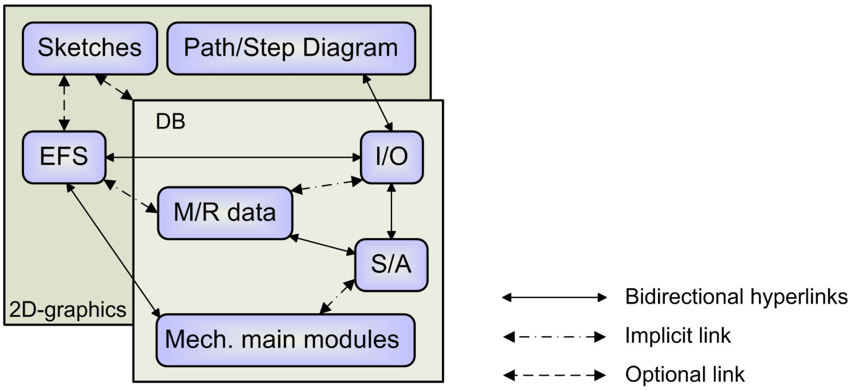


Figure 6. Linkage of the interdisciplinary concept data

The bidirectional hyperlinks have to be defined by the interdisciplinary engineering team. The consistency due to changes in the concept is ensured by a C# Visio add-on.

2.2 Transition from the system design to the domain specific design (II)

It is not only important to connect the domains but also the design phases when establishing an integrated procedure for an interdisciplinary design team. According to [1] the interdisciplinary concept is a base for the domain-specific design, which consists of CAD modeling and PLC software development in the case discussed in this work (Figure 4). The task is now to extract the relevant data from the concept for each domain. A new predefined process module (Figure 7) is developed for the transition from the system design phase to the domain-specific design phase.

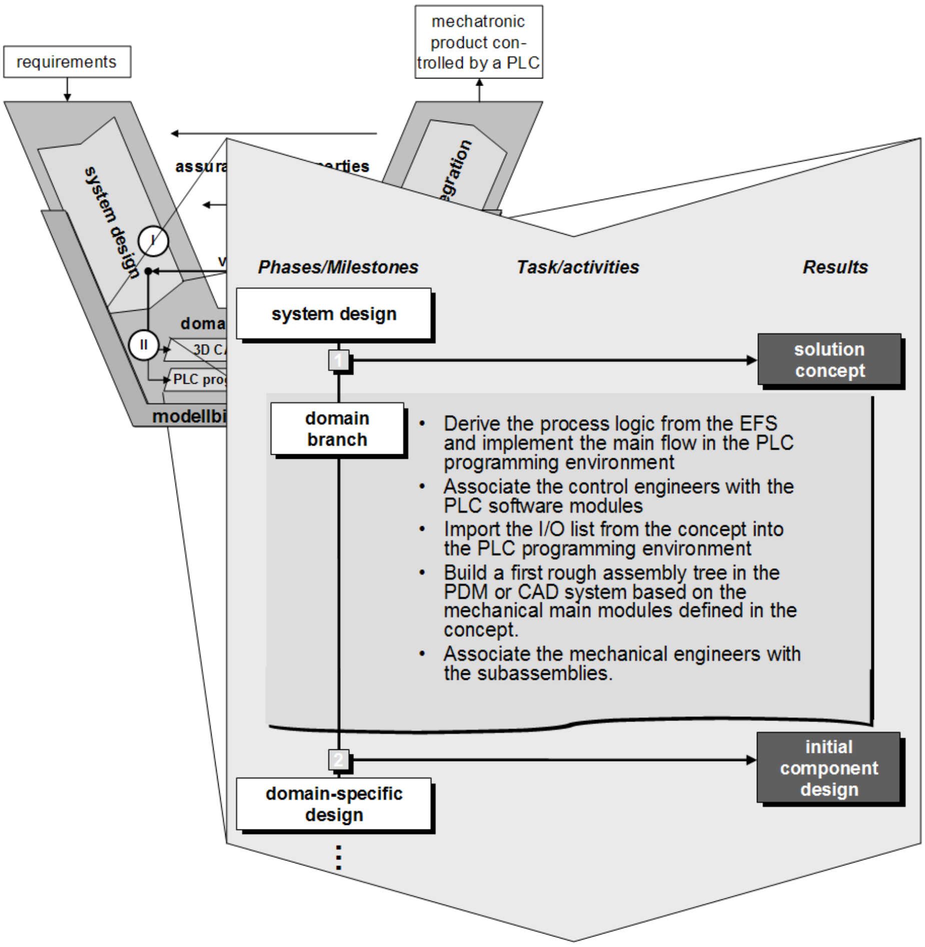

requirements
mechatronic product controlled by a PLC
system design
assurance of properties
integration
I
II
domain
3D CAD
PLC programming
modellbildung
Phases/Milestones
Task/activities
Results
system design
solution concept
domain branch
• Derive the process logic from the EFS and implement the main flow in the PLC programming environment
• Associate the control engineers with the PLC software modules
• Import the I/O list from the concept into the PLC programming environment
• Build a first rough assembly tree in the PDM or CAD system based on the mechanical main modules defined in the concept.
• Associate the mechanical engineers with the subassemblies.
1
2
initial component design
domain-specific design

Figure 7. Predefined process module for the domain branch

Derive the process logic from the EFS and implement the main flow in the PLC programming environment: The main process logic is inherent in the EFS. The method presented in [2] – neglecting the pure mechanical functions in the function structure – allows the systematic derivation of the logic used in the control. This logic can be expressed in all PLC programming languages standardized in IEC 1131-3 [13]. SFC (Sequential Function Chart) is chosen since this language is intended to initiate the PLC programming process [4]. If a digital representation of the concept is available an automatic derivation of SFC is possible, which together with the I/O list leads to a first compileable control software.

Associate the control engineers with the PLC software modules: To initiate the concurrent engineering inside of the software domain software modules are defined [4] based on the initial SFC program [6]. The available control engineers are associated with those modules. The control software can then be written in SFC or in one of the other four IEC1131 languages in combination with SFC [4]/[13].

Import the I/O list from the concept into the PLC programming environment: The input and output variables stored in the I/O list are representing the sensor and actor interface for the control. This I/O list is part of the interdisciplinary description language described in section 2.1. Importing this list into the PLC programming environment enables, together with SFC, the initial version of the control software executable.

Build a first rough assembly tree in the PDM or CAD system based on the mechanical main modules defined in the concept: CAD is the tool in mechanics for the domain-specific design. A PDM system is in many cases supporting the data management. To initiate this task a first rough assembly tree containing the mechanical main modules is defined [6] in the CAD system or – if available – in the PDM system. The concept defined according to section 2.1 does already provide information about the mechanical main modules und the sensors and actors needed in the corresponding module.

Associate the mechanical engineers with the subassemblies: Analog to the concurrent engineering inside of the software domain the subassemblies defined in CAD or PDM are associated with the mechanical engineers.

2.3 Modeling and model analysis (III)

As a support to the development process of mechatronic systems, model analysis is often used. The models are developed with the intention of answering questions arising during the product development process. The models can either be virtual or physical depending on the system. In the VDI guideline 2206 [1] there is a general description of many aspects concerning modeling and model-based system design. In this work the focus is on machines controlled by a PLC. A generic framework is developed to enable the modeling and model analysis for machines in general [9]/[11]/[14]. This framework has been proven in several case studies; see for example [14]/[20]/[21]. The main components are schematically shown in Figure 8.

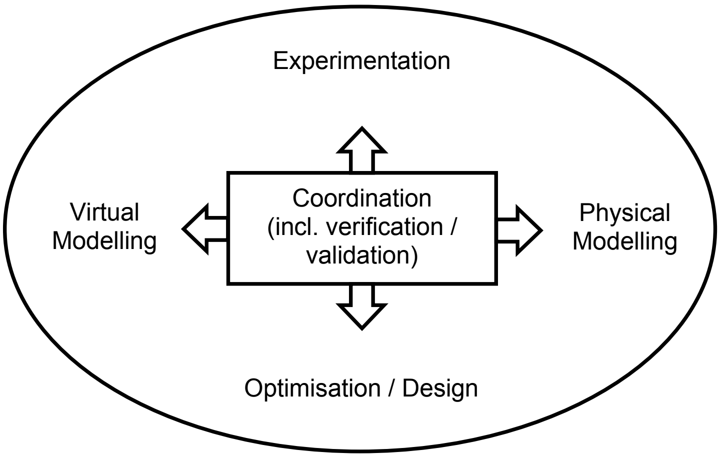


Figure 8. Coordinated approach for model development

In short, virtual models for description of interesting product characteristics are developed, verified and used for initial experimentation (simulation). The simulation results are compared to results from limited physical models developed in parallel, or to experience from previous development projects, for the purpose of validation of the virtual models. Coordination also means that the virtual models are used to design good physical models and measurements strategies. This process is iterated until there is good agreement and the initial question from product development is answered.

2.4 Domain-specific design (IV)

The partitioning [1], dividing the performance of the function among the domains involved, is already described in section 2.2. The resulting linkage shown in Figure 9 can be used to notify the respective other domain in case of changes. A change in the CAD model affecting the sensors or actors has an instant impact on the input and output variables which again affects the PLC program. A change in the PLC program affecting the I/O list has an analog consequence for the CAD development.

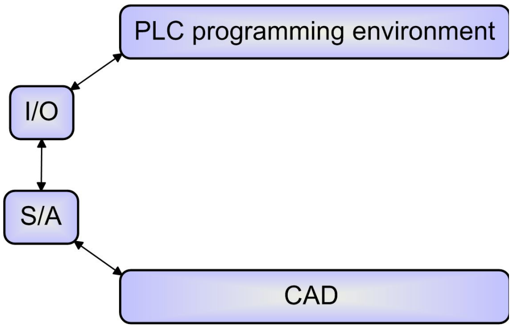


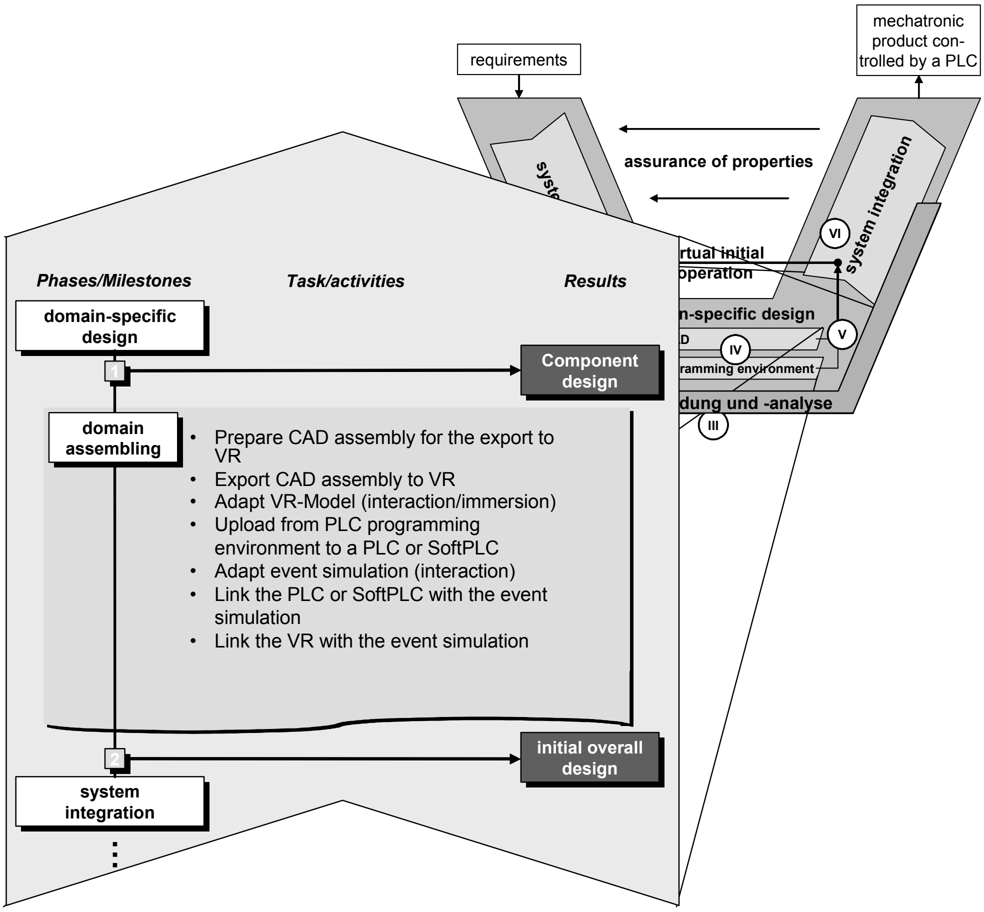
Figure 9. Linkage of the domain-specific design data

As described in [1] the development in the relevant domains takes place on the basis of established, domain-specific development methodologies. The VDI guideline 2221 [5] and the methodology described in [3] is supporting the mechanical engineering.

Beside the references for software engineering in [1] the IEC 1131 standard in combination with [4] supports the PLC software development. In addition to the VDI guideline 2206, the linkage possibilities between the two discussed domains during the domain specific design are shown in this section. Based on the data generated and linked during the domain specific design the transition to the system integration is described in the next section.

2.5 Transition from the domain-specific design to the system integration (V)

A new predefined process module is introduced to describe the transition from the mechanical and the software design into an initial overall design as shown in Figure 10.



- Prepare CAD assembly for the export to VR
- Export CAD assembly to VR
- Adapt VR-Model (interaction/immersion)
- Upload from PLC programming environment to a PLC or SoftPLC
- Adapt event simulation (interaction)
- Link the PLC or SoftPLC with the event simulation
- Link the VR with the event simulation

Figure 10. Predefined process module for the domain fusion

This initial overall design prepares the virtual initial operation explained in the next section to assure the properties defined in the machine concept (section 2.1). The proposed tasks/activities for the domain assembling are described in this section:

Prepare CAD assembly for the export to VR: In order to enable an immersive and fast 3D visualization for the setup shown in Figure 1 three CAD tasks are proposed in this work:

1. An object filter can suppress all unnecessary geometry elements such as screws and nuts.
2. A feature element filter reduces geometric details such as rounding or holes which are not relevant in the VR model.
3. The tessellation accuracy for the model export has to be defined. Hereby all curved geometries are approximated by triangles. The user has to define an accuracy that still represents the objects in a nice way but minimizing the amount of triangles/complexity at the same time.

These three steps increase the performance of the VR model significantly.

Export CAD assembly to VR: The preprocessed CAD model is exported to visualize the machine in a VR tool (Figure 1). Two file formats are mainly used for that purpose: VRML (Virtual Reality Modeling Language) and JT (Jupiter Tessellation).

Adapt VR-Model: In order to reflect the kinematic behavior of the machine rigidly linked modules must be grouped and joints have to be defined. Optional enhancements of the VR model such as texturing, lightning, etc. are possible to improve the appearance of the machine.

Upload from PLC programming environment to a PLC or SoftPLC:

The concept illustrated in Figure 1 relies on a real control. A so called SoftPLC, where the control software is executed on a Personal Computer, can also be used. The compiled software must in both cases be uploaded to the control. The upload includes the hardware configuration, the code, the entire I/O-list as well as its memory addresses.

Export S/A simulation models and I/O-list to the event simulation: For the interdisciplinary initial overall design, the two parts VR and PLC/SoftPLC are connected via a machine simulation as shown in Figure 1. Since the control is a PLC in this work an event simulation processes the signals from and to the control as shown in Figure 11. The focus in this setup is not the profound simulation of the machine – which already is treated in section 2.3 – but the consistency check between the two domains described in the next section.

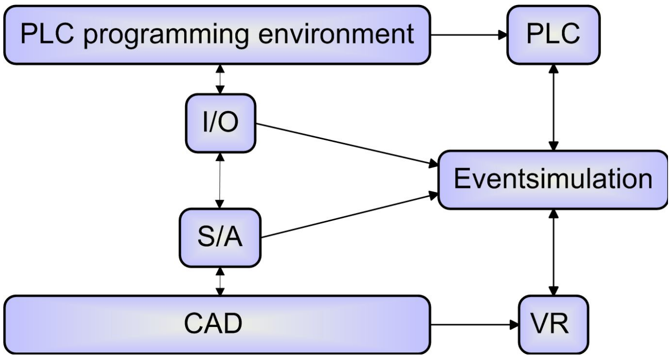


Figure 11. Deriving the virtual machine

The I/O list can be exported from the PLC programming environment and imported in an event simulation such as WinMOD or SIMIT to establish the connection to the PLC. Predefined sensor/actor models (macros) can be used in the event simulation to translate the control signals into geometrical values for the 3D visualization. Especially the actor signal must be translated into a translation or rotation to trigger the kinematics in the VR.

Adapt event simulation: Analog to the refinement of the VR model, the event simulation must be supplemented. Depending on the machine, further dependences between actors and sensors have to be defined [10].

Link the PLC or SoftPLC with the event simulation: Finally all data are now prepared and must only be connected among each other. The bidirectional connection between the PLC/SoftPLC and the event simulation is usually established by the use of a field bus [10].

Link the VR with the event simulation: All VR tools used so far by the authors ([7], [15], [16] and [17]) can be linked to an event simulation via TCP/IP. At least the translation and the rotation values have to be passed to the visualization. Additional information such as collisions can also be transmitted [10].

2.6 Virtual initial operation (VI) in the system integration

The virtual machine [10]/[11]/[9]/[18], illustrated in Figure 1 and derived in the last section, is initializing the system integration. The CAD model from the mechanical engineers is now connected to the software developed by the control engineers. All three types of integration mentioned in the guideline VDI 2206 [1] (modular, spatial and the integration of distributed components) are covered. The modular and spatial integration is virtually realized in the CAD model, the simulation model and the PLC software. Field bus communication or direct API (Application Programming Interface) access is used to connect the real control.

In opposition to the VDI guideline 2206 [1] the morphological box is not recommended by the authors to check the incompatibilities during the system integration. The functional consistency is ensured due to the virtual machine. Conformable to [3] the morphological box should already be used in the system design phase to choose an overall solution. This solution variant consists of the selected partial solutions for the functions defined in the EFS as proposed in section 2.1.

The first assurance of properties (Figure 3), realized by the virtual initial operation (Figure 4), is validating the machine concept described and parameterized in section 2.1. All functions, modules, actuators, sensors, steps and machine/recipe data can be validated by the use of the virtual machine. This assurance step is proposed in this work before deepening the system integration according to [1]. A case study will exemplify the presented adaptations of the VDI guideline 2206 [1] in the following chapter.

3. Case Study

Various industry partners are working together with the authors on the topics described in this work. A typical mechatronic system controlled by a PLC is a textile machine from “RIETER textile” [8]. A comber from RIETER has been analyzed and specified in Visio using the interdisciplinary description language from section 2.1. A cutout of the extended function structure for that comber is shown in Figure 12.

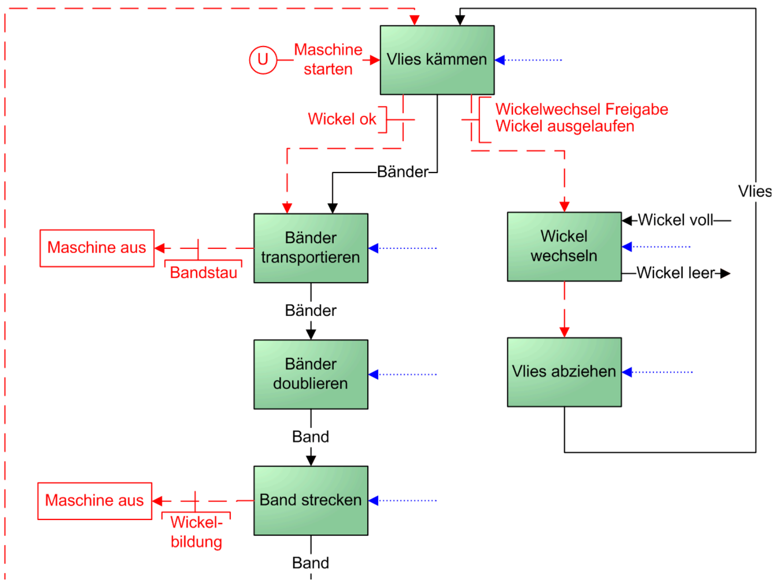
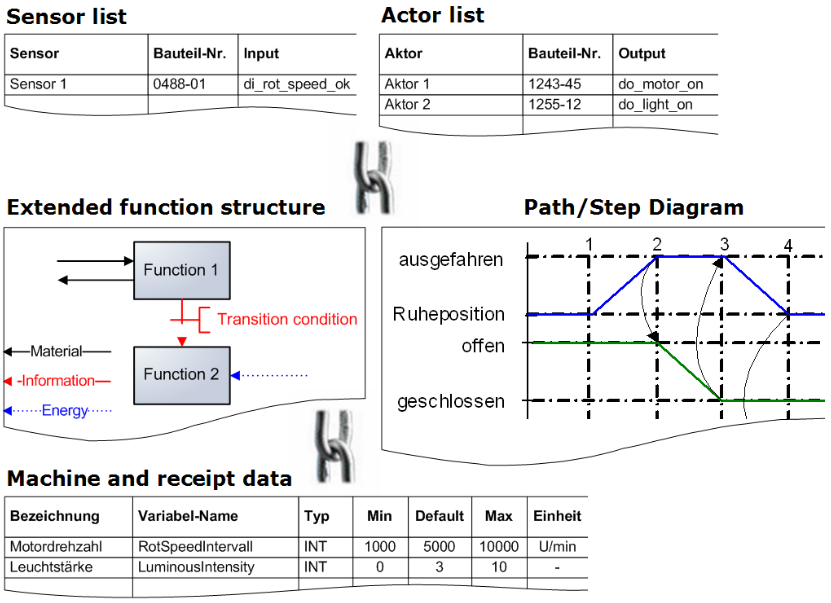


Figure 12. Cutout of the EFS for the comber in Visio

The information flow (dashed red) is the base for the control software depict in Figure 14. Sub functions are recursively defined and linked. Bidirectional hyperlinks are also defined in-between the other elements used in the interdisciplinary description language as shown in Figure 7 and exemplified in Figure 13.



Sensor	Bauteil-Nr.	Input
Sensor 1	0488-01	di_rot_speed_ok

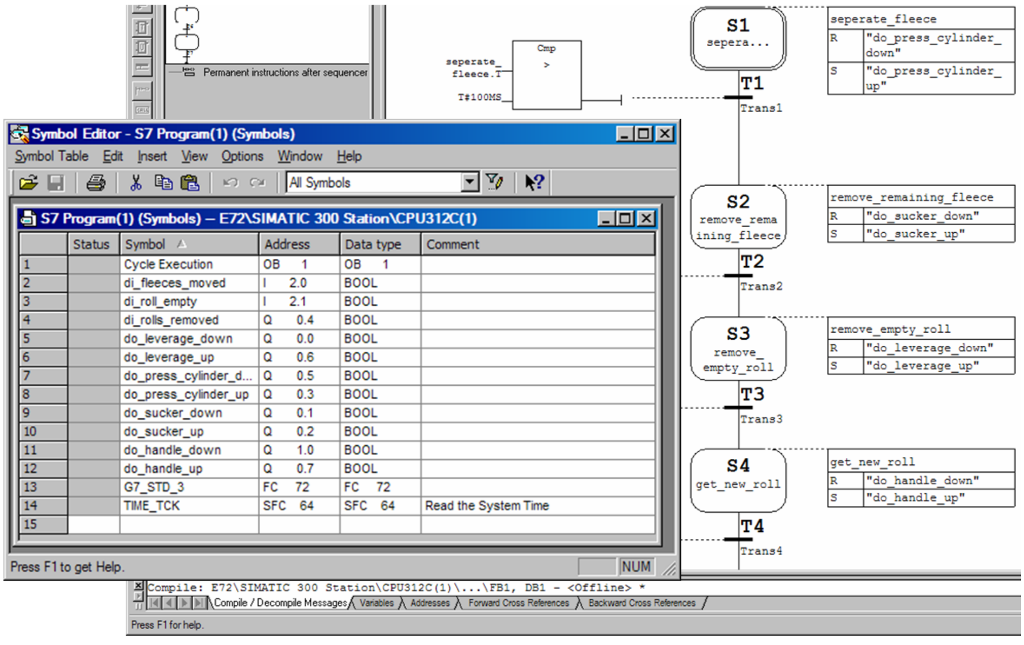
Aktor	Bauteil-Nr.	Output
Aktor 1	1243-45	do_motor_on
Aktor 2	1255-12	do_light_on

Bezeichnung	Variabel-Name	Typ	Min	Default	Max	Einheit
Motordrehzahl	RotSpeedIntervall	INT	1000	5000	10000	U/min
Leuchtstärke	LuminousIntensity	INT	0	3	10	-

Figure 13. Linked description elements for the comber example in Visio

As an example, a transition condition in the EFS contains a link to its input variable in the I/O list, which again is connected to a sensor. Web pages exported from Visio are useful to browse the concept in read only mode.

A section of the resulting PLC programming language SFC according to the predefined process module defined in section 2.2 is shown in the next figure.



	Status	Symbol	Address	Data type	Comment
1		Cycle Execution	OB 1	OB 1	
2		di_fleeces_moved	I 2.0	BOOL	
3		di_roll_empty	I 2.1	BOOL	
4		di_rolls_removed	Q 0.4	BOOL	
5		do_leverage_down	Q 0.0	BOOL	
6		do_leverage_up	Q 0.6	BOOL	
7		do_press_cylinder_d...	Q 0.5	BOOL	
8		do_press_cylinder_up	Q 0.3	BOOL	
9		do_sucker_down	Q 0.1	BOOL	
10		do_sucker_up	Q 0.2	BOOL	
11		do_handle_down	Q 1.0	BOOL	
12		do_handle_up	Q 0.7	BOOL	
13		G7_STD_3	FC 72	FC 72	
14		TIME_TCK	SFC 64	SFC 64	Read the System Time
15					

seperate_fleece	
R	"do_press_cylinder_down"
S	"do_press_cylinder_up"

remove_remaining_fleece	
R	"do_sucker_down"
S	"do_sucker_up"

remove_empty_roll	
R	"do_leverage_down"
S	"do_leverage_up"

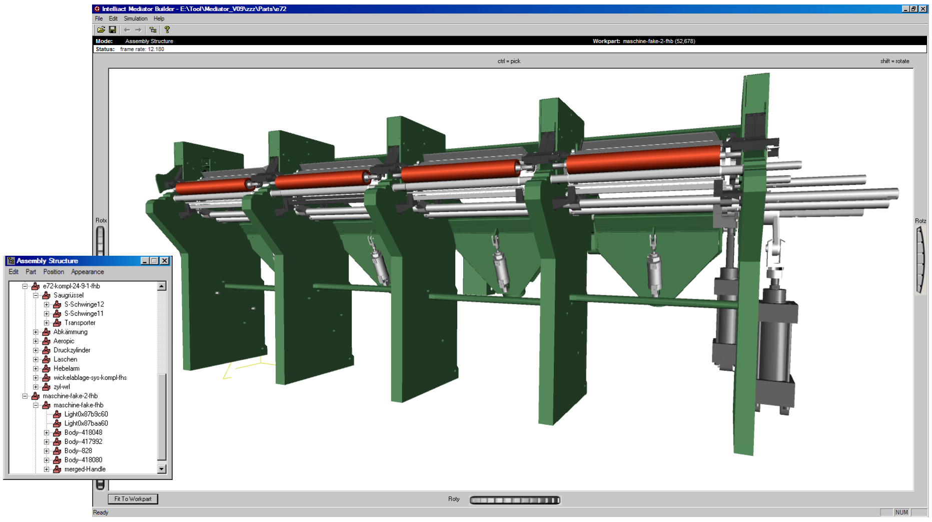
get_new_roll	
R	"do_handle_down"
S	"do_handle_up"

```
Compile: E72\SIMATIC 300 Station\CPU312C(1)\...\FB1, DB1 - <Offline> *
```

Figure 14. SFC and the I/O list in the PLC programming environment

All I/O's used, linked and listed in the concept are in addition to the SFC imported in the PLC programming environment Figure 14. This first software version can be compiled and the debugging feature allows stepping through the sequence by simulating sensor signals via the input variables. The software development based on this software skeleton is described in [4].

RIETER is practicing the top down approach in CAD modeling: First of all, the main modules are defined in a PLM system and human resources are associated. The initial rough assembly tree is completed in conjunction with the geometry modeled in the CAD system. The resulting CAD model and its assembly tree are visualized in Figure 15 using the software Mediator from Intelliact [7].



Mode: Assembly Structure
Workpart: maschine-fake-2-fhb (52,678)
Status: frame rate: 12.180

- e72-kompl-24-9-1-fhb
  - Saugrüssel
    - S-Schwinge12
    - S-Schwinge11
    - Transporter
  - Abkämmung
  - Aeropic
  - Druckzylinder
  - Laschen
  - Hebelarm
  - wickelablage-sys-kompl-fhs
  - zyl-wrl
- maschine-fake-2-fhb
  - maschine-fake-fhb
    - Light0x87b9c60
    - Light0x87baa60
    - Body--418048
    - Body--417992
    - Body--828
    - Body--418080
    - merged-Handle

Figure 15. Geometry and assembly tree of the comber

RIETER used WinMOD in this case study to simulate the events between their B&R control [12] and the comber (section 2.5). The additional bi-directional connection between WinMOD and the VR software Mediator leads to the virtual machine setup illustrated in Figure 16 for the comber.

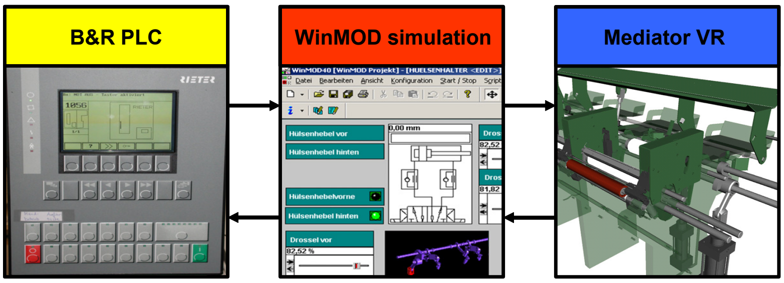


Figure 16. The virtual machine for the comber

Beside the advantages during the development process RIETER realized typical additional advantages in other processes of the enterprise such as service, marketing and education.

4. Conclusion

The customized guideline presented in this work is an adaptation of the German VDI-guideline 2206. This work focus on development of machines controlled by a PLC and the study does not claim to be able to handle every kind of mechatronic system. Developments viewed in isolation by earlier studies done by the authors are here considered together. All aspects from the VDI guideline 2206 are taken into account such as: verification, validation, hard-ware-in-the-loop (HIL), concurrent interdisciplinary engineering, etc. The methods and tools developed enable a more efficient development process for machines controlled by a PLC. To verify the relevance and functionality of the adapted guideline, a case study is performed showing the potential for an improved product development process using the enhanced methodology presented in this paper. However, the data flow needs to be further detailed and automated to be more efficient.

Acknowledgments

Financial support from the Swiss innovation promotion agency CTI (KTI), Rieter Textile Systems AG, Intelliact AG, GRITEC Institut für angewandte Technologie AG and BRÜTSCH Elektronik AG is gratefully acknowledged.

References

[1] "VDI 2206: Design methodology for mechatronic systems", Beuth Verlag GmbH, 10772-Berlin, Düsseldorf, Germany, June 2004.

[2] Bathelt J., Jönsson A., Bacs C., Kunz, A, Meier, M., "Conceptual design approach for mechatronic systems controlled by a programmable logic controller (PLC)", ICED03 International Conference on Engineering Design, Stockholm, Sweden, 2003.

[3] Pahl G. and Beitz W., "Engineering design: a systematic approach", Springer, Berlin, 1999.

[4] Bonfatti F., Monari P. D. and Sampieri U., "IEC 1131-3 Programming Methodology", CJ International, France, 1997.

[5] "Systematic Approach to the Design of Technical Systems and Products", VDI Design Handbook 2221, VDI-Verlag, Düsseldorf, Germany, 1987.

[6] Bathelt J., Jönsson A., Bacs C. and Meier M., "Modularisierung SPS-gesteuerter mechatronischer Systeme", Proceedings of the 1st Paderborner Workshop: Intelligente mechatronische Systeme, Paderborn, Germany, 2003.

[7] “Intelliact Mediator”, http://www.intelliact.ch/home.php?id=100034, Zurich, Switzerland, visited 2005.

[8] “Rieter Textile Systems”, http://www.rieter.com/main/textile/, Winterthur, Switzerland, visited 2005.

[9] Bathelt J. and Jönsson A., "How to Implement the Virtual Machine Concept Using xPC Target", Nordic MATLAB Conference, Copenhagen, Danmark, 2003.

[10] Dierssen S., “Systemkopplung zur komponentenorientierten Simulation digitaler Produkte”, VDI-Verlag / Reihe 20 / Nr. 358, Germany, 2002.

[11] Jönsson A., “Lean Prototyping of Multi-body and Mechatronic Systems”, Dissertation Series No. 2004:08, Blekinge Institute of Technology, Karlskrona, Sweden, 2004.

[12] “Bernecker + Rainer Industrie Elektronik GmbH”, http://www.br-automation.com/, Eggelsberg, Austria, visited 2005.

[13] “IEC 1131-3: Programmable controllers – Part 3: Programming languages”, Beuth Verlag GmbH, Berlin, Germany, 1994.

[14] Jönsson A., Wall J. and Broman G., “A virtual machine concept for real-time simulation of machine tool dynamics”, International Journal of Machine Tools and Manufacture, 45, 795-801, 2005.

[15] „VirtualDesign2”, http://www.vrcom-online.de/en/product/index.html, Darmstadt, Germany, visited 2005.

[16] “Virtual Reality Toolbox”, http://www.mathworks.com/products/virtualreality/, Natick, Massachusetts, United States of America, visited 2005.

[17] “EON”, http://www.eonreality.com/, Alton, United States of America, visited 2005.

[18] Kreusch K., “Verifikation numerischer Steuerungen an virtuellen Werkzeugmaschinen”, Shaker Verlag, Aachen, Germany, 2002.

[19] Otto, K. N., “A Process for Modularizing Product Families”, ICED01 International Conference on Engineering Design, Glasgow, 2001.

[20] Jönsson A. and Broman G., “Experimental investigation of a rammer soil compactor machine”, Proceedings of the NAFEMS World Congress, Como, 2001.

[21] Wall J., “Dynamic Study of an Automobile Exhaust System”, Licentiate Dissertation Series No. 2003:08, Blekinge Institute of Technology, Karlskrona, Sweden, 2003.

Jens Bathelt, Christian Bacs, Stefan Dierssen, Prof. Dr. Markus Meier
Federal Institute of Technology, Institute of Mechanics, CH-8092 Zurich, Switzerland
Tel: +41 44 632 35 15, Fax: +41 44 632 11 81, E-mail: bathelt@imes.mavt.ethz.ch
URL: http://www.zpe.ethz.ch

Anders Jönsson
Blekinge Institute of Technology, Department of Mechanical Engineering, SE-371 79 Karlskrona, Sweden
Tel: +46 455 38 55 12, Fax: +46 455 38 55 07, E-mail: anders.jonsson@bth.se
URL: http://www.bth.se/tek/amt