INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

NEW DATAHANDLING- AND RAPID VIRTUAL PROTOTYPING APPROACH FOR MECHATRONIC SYSTEMS

Bacs Christian[1]
[1]ETH Zürich, ASL, Switzerland

ABSTRACT
Simulating mechatronic systems using a virtual prototype is important since many errors and inconsistencies in soft- and hardware can be detected without the need of expansive and long winded prototypes. Thus, together with an increase of quality, time and money can be saved. However, examples in industry show that these advantages are reduced enormously by data inconsistency or even no existing data flow at all. This paper describes an approach for rapid virtual prototyping of manufacturing systems. Present data of a CAD-system are connected to mechatronic standard components (MSC), which in turn can be exploited to generate the virtual prototype consisting of an event based simulation, its real control and 3D-visualization. Furthermore, advantages of this approach during the domain-specific design are pointed out. Finally, a case study reflects this approach within an industrial frame.

Keywords: Mechatronic Product Development, Rapid Virtual Prototyping (RVP), Mechatronic Standard Component (MSC), Virtual Machine, Information Flow, Data Consistency, VDI Guideline 2206

1 INTRODUCTION
Mechatronic is a new innovation source of superior products. It combines closely the three domains of mechanics, electronics and information processing. Mechatronics aims at controlling the basic system using an information flow [1]. Thus, new solutions in mechanical engineering can be made. These can improve the rate between costs and value of known products as well as stimulate completely new products. To verify such complex products the concept of the virtual machine [2], [3], [5], [6], [7] can be used before setting up an expansive prototype. Based on pure digital information, the virtual machine simulates a mechatronic system connected to its real control, which is linked with an event based simulation as shown in Figure 1. The simulation reads synchronously with the communication cycles of the control the states of the actuators and calculates consequentially the state of the sensors. These states are then transmitted back to the real control and/or used for visualization.

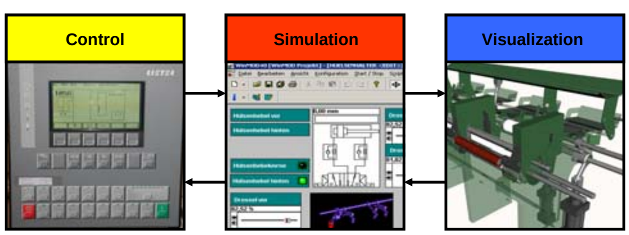


Figure 1 Concept of a virtual machine

Unfortunately, the concept of the virtual machine remains an isolated application as long it is not integrated into the digital development process. Furthermore, the collection of the relevant data needed to build up a virtual machine is long winded since the interdisciplinary design proceeds in different domains resulting in widespread data storage. Although most of these relevant data is already stored centrally on a server, such as a product lifecycle management system (PLM), they are not exploited. Especially the machine simulation is still made by hand, which causes version conflicts and redundancy between the current data and the simulation model. This Paper aims at developing an

approach and a guideline for rapid virtual prototyping of mechatronic systems in such a way as current data can be exploited to generate the virtual machine and be used within applications of the domain specific design.
The V-Model of the VDI 2206 Guideline for the development of mechatronic systems [8] in Figure 2 gives an overview of the critical points to be solved by this approach.

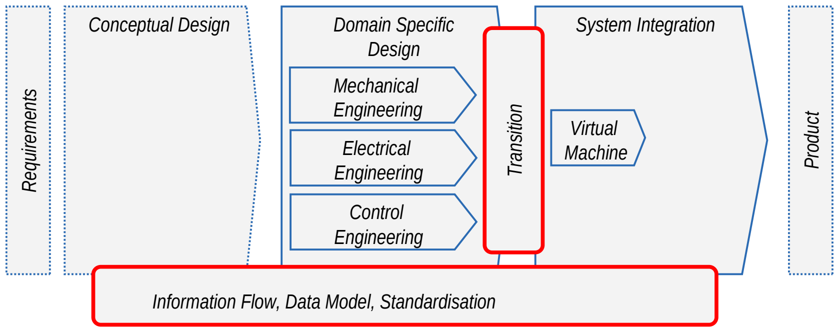


Figure 2 Critical points within the V-Model of the VDI 2206 Guideline

The V-Model provides a general frame for the design of any kind of mechatronic systems from the requirements up to the real product. The problem-solving circle consists of the three modules *"Conceptual Design"*, *"Domain Specific Design"* and *"System Integration"*, whereas each of them describes a predictable subtask in the development of mechatronic systems. The results made within these subtasks must be continually checked on the basis of the specified requirements and the solution concept. Therefore, it must be ensured that the actual system properties coincide with the desired specifications. Overall, *"Modelling and Model Analysis"* investigates the system properties using models and computer-aided tools for simulation such as the concept of a virtual machine. On the basis of this V-Model, the suggested approach will be methodically integrated to ensure consistency with the current standard. There are two crucial points: the approach must ensure the interdisciplinary information flow and the transition to the virtual machine, the rapid virtual prototyping (RVP).

2 APPROACH

The approach is based on exploiting mechatronic components, which reoccur consistently during the development process and hence are suitable for standardisation. These components consist of basic and interdisciplinary used data and are merged in subcomponents. A subcomponent could cover e.g. the behaviour within the event based simulation or an simple shape used in a pneumatic scheme. With such mechatronic standard components (MSC), several advantages can be achieved. Since these components are stored locally on a PLM, all development domains can access them and work now interdisciplinary on a common dataset. Thus, data redundancy is decreased as well as data consistency is ensured over all domains. Especially the last mentioned is important in cases of elementary changes during the development process. Additionally, the interrelationship of the widespread data is now transparent via the MSC's since interdisciplinary data are now linked digitally to each other on the PLM compared to nowadays where interdisciplinary data are still linked analog e.g. over an ID within documents.
The principal setup of a mechatronic system as shown in Figure 3 leads to a possible composition of such mechatronic standard components.

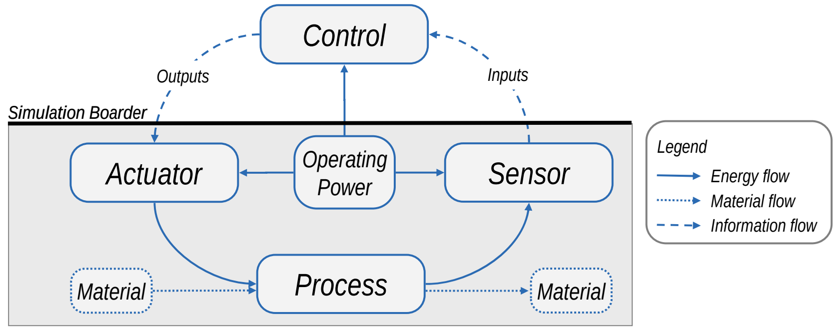


Figure 3 Principal setup of a mechatronic system

A mechatronic system, in this case simulated by a virtual machine, consists of the four elements: a control unit, actuators, sensors and a process, which represents the pure mechanic system and the logical linkage between the outputs and the inputs of the control [4]. The cycle starts usually at the control unit, e.g. by pushing a start-button. Commands for the actuators are calculated and transmitted as output signals over a BUS-system. These feed energy into the process, e.g. as a movement, temperature etc. Within the process, material such as work pieces will be machined. To regulate the process, sensors observe the machine's states and forward them as input signals back to the control. Herewith, the cycle starts again by recalculating new commands for the actuators. Using the concept of a virtual machine, actuators, sensors and the process are simulated while the control remains real (hardware in the loop). This is indicated by the simulation boarder in Figure 3. Obviously, actuators and sensors are the pivotal point of the three mechatronic domains mechanical-, control- and electrical-engineering and are therefore candidates for the MSC's. In addition, they are preferably outsourced items in industry and thus promise to be standardized. A survey in industry [5] shows in the following Table 1 the most frequently used simulation components within mechatronic systems.

Pneumatics/Hydraulics		**Sensors**	
single action cylinder	21%	mechanical switch	6%
double action cylinder	27%	proximity sensor	25%
valve	23%	pressure switch	10%
positive displacement pump	12%	absolute measurement system	23%
other	17%	other	36%
Electrics		**Kinematics**	
unregulated Motors	31%	translative axes	49%
NC-axes	45%	rotative axes	23%
other	24%	gears	12%
		other	16%

Table 1 Components in a simulation

At first view, the 4 components in *Pneumatics/Hydraulics*, the 4 components in *Sensors*, the 2 components in *Electrics* and the 3 components in *Kinematics* cover approximately 75% of all used simulation components. However, each component may have numerous variations. In order to deal with this variety, Chapter 3.1 will show their behaviour within the event based simulation and thus how easily they can be categorised into a few standard components (Macros).

The overall idea of this approach, using actuators and sensors as MSC's and thus providing a platform for all domains and the virtual machine, is illustrated in Figure 4. On the bottom level, a cylinder illustrates a possible MSC. With other MSC's, assemblies can be built up, which in turn can be defined as a MSC again. Carried on like this, machines and plants can be assembled, copied and connected to each other. Analogue thoughts were described by [9] and [10].

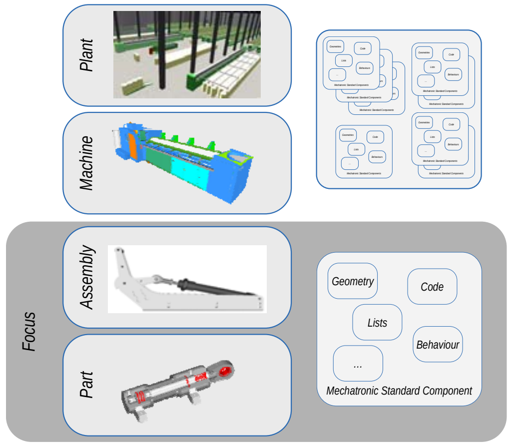


Figure 4 Overall idea and focus

MSC's may not only be used to build up event based simulations. An important benefit can also be found during the domain specific design, which shows up promising parallels using such components in different domains, as illustrated in Figure 5. Mechanical engineers already use standard elements in CAD. Nowadays, a PLM system disposes a limited choice of actuators and sensors with different varieties and properties. The linkage of these standard elements, which are primarily defined by the mechanical engineer, with the corresponding MSC's could provide a pivotal platform for all relevant data, which now can be used by all other domains and the virtual machine.

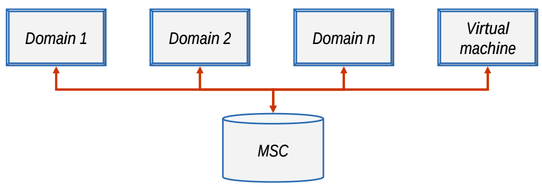


Figure 5 Usage of MSC's

Chapter 4 focuses on the further usage of mechatronic standard components within domains by using this approach.

3 USING THE APPROACH FOR A VIRTUAL MACHINE

3.1 Event based simulation

The event based simulation is the core of a virtual machine and represents the logical behaviour of the mechatronic system. The event based simulation reads the output signals for the actuators and recalculates on this basis in real-time the input signals of the sensors. A signal is thereby one bit if it is digital (True/False) or 4 bytes long if it is analog (Value). The input signals are finally sent back to the real control while the states of the actuators are needed for the visualisation. Thus, an event based simulation represents the logical behaviour between actuators and sensors. Attributes such as loads, acceleration or material behaviours are not taken account of event based simulations compared to dynamic simulations.
To support the rapid virtual prototyping, the event based simulation must consist out of the complete input/output-list (I/O-list) and out of the macros of the equivalent MSC's. While nowadays the placement and assembling of these macros and the I/O-list is done manually, the configured MSC's would allow calculating the dependencies between the macros and the I/O-list and placing and assembling them from the outside. This renewal demands the capability to import data from the MSC's into the simulation tool. Thus, the event based simulation can be set up automatically and rapid virtual prototyping is guaranteed. Typical simulation tools are WinMOD [11] and SIMIT [12], which both support these requirements. WinMOD will be discussed in more detail in the case study in Chapter 4.

The macros within the event based simulation tool are theoretically made up of all possible existing actuators and sensors. Since there is an uncountable amount of different actuators and sensors, a classification was made specifically regarding their behaviour within the event based simulation. E.g. to a control, a mono- and bistable cylinder behave equally. The communication over I/O's as well as the analog state for the visualization are the same. Only their parameterization and assembling of the I/O's to the right slots must be adapted. Most actuators and sensors have been examined on their behaviour and are summarised in Figure 6 and Figure 7.

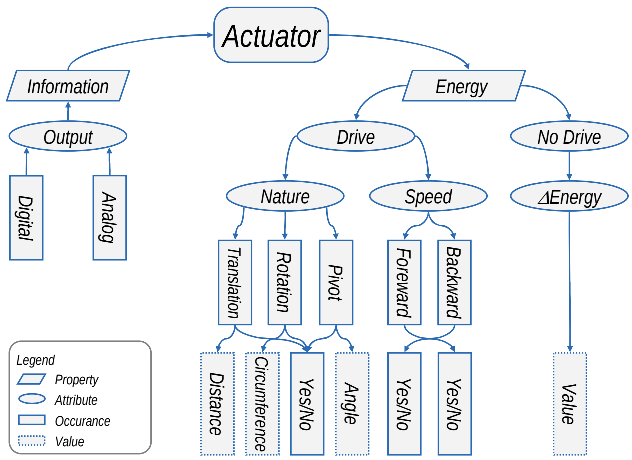


Figure 6 Categorized actuators

An actuator has two properties: information and energy. As it was shown in Figure 3, an actuator receives an information flow from the control and transcribes it by means of operating power to an energy flow. The information received from the control is a digital or an analog signal. This signal initiates the state of the actuator, which results in a drive or any kind of power such as heat. The drive can be differed by its nature and speed. Possible natures are translations, rotations and pivots, whereby rotation is an unlimited and a pivot a limited movement. In addition, they can run forward or backward. This categorization leads to the macros, which are without values, since these will be defined from external by the MSC's.

Sensors are much easier to deal with. They are categorized as shown in Figure 7.

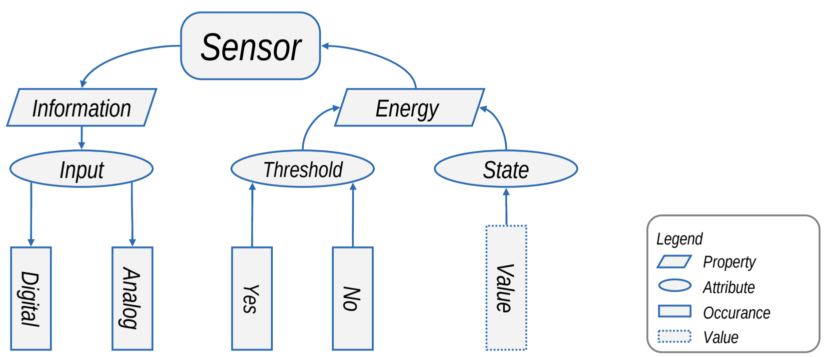


Figure 7 Categorized sensors

Sensors consist, just as actuators, of the two properties information and energy. This time, as it has been shown in Figure 3, a sensor receives an energy flow from the process and transcribes it by means of operating power to a digital or analog signal. These input signals are sent back to the control. The input signal is initiated either by reaching a threshold, e.g. a certain temperature, which results in a digital input, or by continuous measurement of the state, which results in an analog signal, e.g. the value of the temperature. Equal to the actuators, sensors are selected and parameterized from external.

3.2 Kinematic loops and visualisation

The visualisation of complex movements within a virtual machine needs kinematic loops as illustrated in Figure 8. In closed loops as they are used in mechatronic systems, link angles and translational displacement of fixed bodies are directly connected and therefore geometrically dependent on each other. Thus, the calculation of all bodies' positions within a closed mechanism consists of several equations. These can be solved by a given movement such as a rotation or a translation of an actuator.

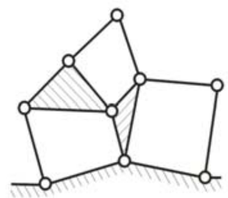


Figure 8 Closed kinematic loop

Almost all kinematic loops in mechatronic systems proceed two-dimensionally. Since the variety of possible loops is therefore low, the equations can be predefined and simply be solved by an analog value of the event based simulation and the allocation of the dependant bodies, axes and alignment. In order to build up a virtual machine, only the connections between these loops and the corresponding actuators must be made within the MSC's. The virtual bodies still need to be allocated manually. In case of changes, the loops will be adapted synchronously by the MSC's.

4 FURTHER USAGE OF MECHATRONIC STANDARD COMPONENTS

Mechatronic standard components may not only be used when initialising a virtual machine. It has been shown that in other domains such as ECAD (Electronic CAD) and hydraulics/pneumatics, actuators and sensors play decisive roles, as descript in the next two chapters. In the last chapter, the overall scheme of the treated topics resumes the approach.

4.1 ECAD

Within the ECAD, all power- and control-related junctions are represented in a scheme. An example is shown in Figure 9. The scheme is divided into a device/component-, cable- and wiring-list. Again, actuators and sensors are handled centrally since they are related to the control and to the operating power scheme. They are attached to clamps, power plugs and the I/O-list from the control: the actuators to the outputs and the sensors to the inputs. Thus, the I/O-list is used in parallel as the wiring scheme.

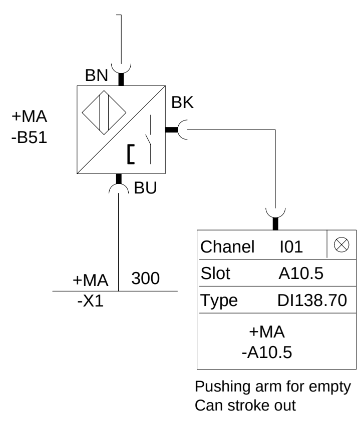


Figure 9 Cylinders in ECAD

A typical characteristic of the ECAD is the identification of all used elements by item designations (ID). For each actuator, sensor, clamp or cable within the scheme, this unique ID is given. Examples from industry show, that this system causes redundancy and inconsistency between domains due to not interconnected domain specific identifications. Thus, an actuator is used and defined several times within the different domains and changes in the mechatronic system are not detected automatically.

Furthermore, most industries like to keep their ID-enumeration for certain elements, meaning that a specific actuator keeps its ID within all schemes. The approach described in Chapter 2 solves these problems, if ECAD is connected to the MSC's.

4.2 Pneumatics and hydraulics

Actuators and sensors running with fluid or gas are similar to ECAD represented in a pneumatic/hydraulic-scheme as illustrated in Figure 10. Up to the present, many tools have been used to develop such schemes. Nowadays, pneumatics- and hydraulics-schemes can also be done within ECAD.

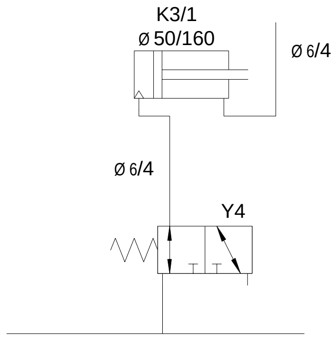


Figure 10 Cylinder in a pneumatic scheme

The scheme presents all pneumatic and hydraulic actuators and sensors including valves, pipes and pressure tank. Properties as pressure or geometrical measurements are appended as well. ID's from ECAD are used. Regarding the approach, all this information can be shared over the MSC's.

5 OVERALL SCHEME

In Figure 11, all covered topics are summarized within the overall scheme. The figure is split into two halves each of them representing one of the critical points defined within the introduction. On the right side, the data handling is illustrated, while on the left side the Rapid Virtual Prototyping is treated. In addition, the figure consists of three main parts: all developing domains such as mechanics on the right, the PLM-System displayed as a database in the centre including the MSC's and the Virtual Machine with its subcomponents *VR*, *Event–Simulation* and *Control* on the left.

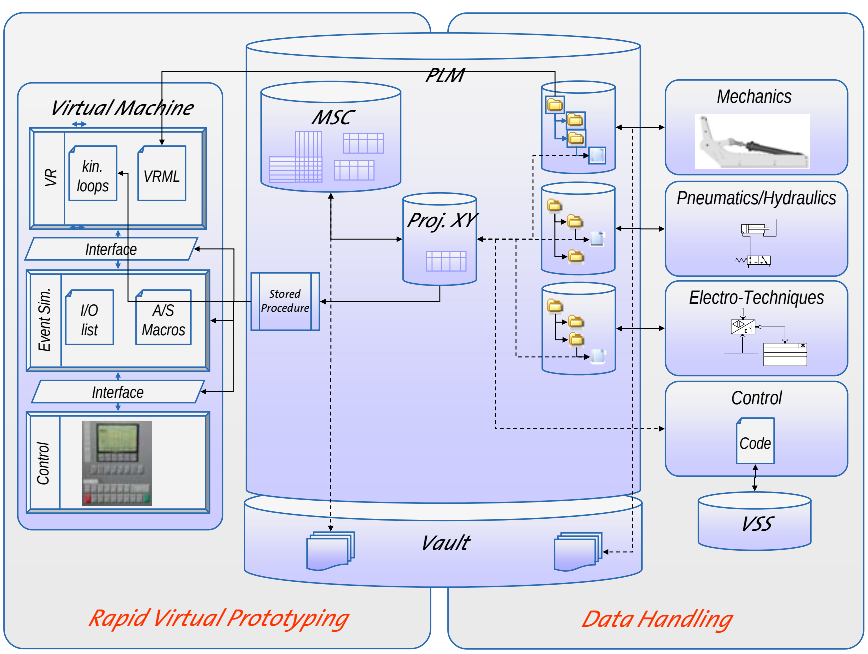


Figure 11 Overall scheme

Typically, mechanical engineers define actuators and sensors within a mechatronic system. Actuators and sensors are chosen by referencing to the MSC's and file an instance to the projects database. The Instances of the MSC's are then parameterized and linked into the domain's specific data-model such as the hierarchy-structure of a CAD-Model. More and more, all disciplines may now complete this instance with their specific data if not already predefined within the MSC. The control-engineer may now fill the IO-list derived from the MSC's. Afterwards he adds the project specific addresses which also can be used by the electrical engineer. The electrical engineer uses the predefined shapes for the electrical-scheme and associates them with the ID's if not already automatically given by the MSC's. Furthermore he receives specifications of the implemented actuators and sensors like the stroke of a cylinder and its addresses to control and monitoring the valve. The list of operations is not completed by far and acts just as examples. With it, all domains are associated with the same not redundant data and in case of changes inconsistency can now be avoided.
These steps are important to initiate the rapid virtual prototyping. Since the MSC's contain all relevant data of all project specific actuators and sensors, the virtual machine may now derived from them. Going through the projects MSC's, the macros within the event-based simulation can be assembled, parameterized and linked with the corresponding inputs and outputs. This outcome is stored in a text file, whereby each line stands for one configuration command. This file is importable and controls the configuration and assembly of the simulation. The VRML model, derived from the CAD-system, is connected with a configuration file, reading the signals from the simulation and calculating the movements of the parts by defining kinematic loops. All these information are derived from the MSC's as well. After the domain specific design is finished, the system integration can be tested by the virtual machine.

6 CASE STUDY

With an industry partner, a case study to this topic was worked out. A cylinder within an assembly of the comber machine from RIETER Textile AG [13] was analyzed and specified in the event based simulation WinMOD and a visualisation tool. The assembly shown in Figure 12 is a closed two-dimensional kinematic loop and is used to move empty cans into the work position.

Figure 12 Cylinder assembly

The assembly was exported of a CAD system on a PLM and imported into the visualization tool as a VRML object. Regarding the approach, the cylinder is defined as a MSC, which is connected to the kinematic loop, the macro of the simulation tool and the I/O list of the real control.

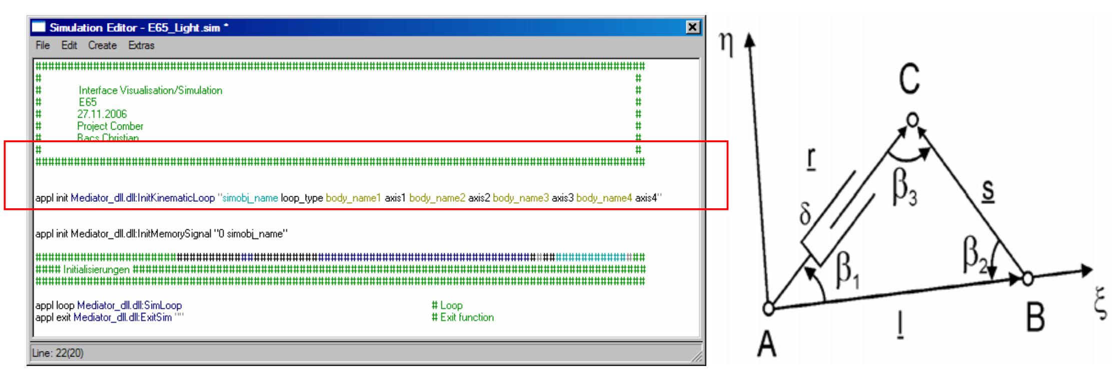


Figure 13 Defining kinematic loops

Behind the visualization, the kinematic loop is defined by the MSC and connected with its bodies and signals from WinMOD. In Figure 13 the text based definition of these loops is shown. In the first line, the type of the kinematic loop is defined and connected with the names of the depending bodies. On the right, the principal geometry of this kinematic loop is shown. The cylinder has the variable δ for its stroke. The other parts are definitely dependant on the stroke and can be computed starting with formula (1).

$$\beta_1 = \pm acos\left(\frac{l^2 + (r+\delta)^2 - s^2}{2l(r+\delta)}\right) \tag{1}$$

The variable δ is connected to the macro within the event based simulation and receives its value over the analog signal placed on the first 4 bytes (0.0 to 3.7).

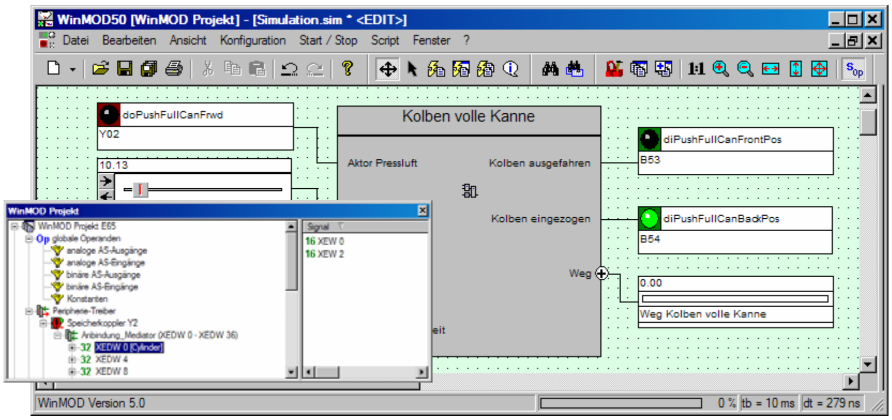
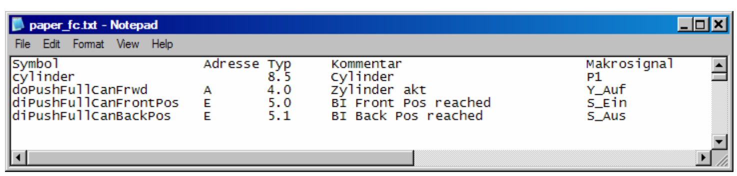


Figure 14 Cylinder in WinMOD

The screenshot in Figure 14 shows a typical macro defined in WinMOD. This macro receives one output signal from the control and calculates the stroke as well as the two input signals 'work position reached' and 'initial position reached'. The outputs and inputs were separately imported from the control. The configuration of this macro and its I/O's is derived from the MSC.


paper_fc.txt - Notepad

Symbol	Adresse	Typ	Kommentar	Makrosignal
cylinder		8.5	cylinder	P1
doPushFullCanFrwd	A	4.0	Zylinder akt	Y_Auf
diPushFullCanFrontPos	E	5.0	BI Front Pos reached	S_Ein
diPushFullCanBackPos	E	5.1	BI Back Pos reached	S_Aus

Figure 15 WinMOD import

Figure 15 shows a text file generated by the MSC, which can finally be imported into WinMOD.

7 CONCLUSIONS

The development of a rapid virtual prototyping approach for mechatronic systems, presented in this work, highlights actuators and sensors as the main spanning components over all domains. Using them as mechatronic standard components, data redundancy as well as error rate can be further reduced and virtual machines can be initiated much faster as a result of all domains working on the same data set. Furthermore, the approach has shown that these components have additional advantages within the domains of electrics and hydraulics/pneumatics. At last, the structure of data is improved significantly, since the complex correlations between interdisciplinary domains can be overviewed. To verify the approach, a case study was performed showing the potential of mechatronic standard components. However, the approach needs further investigations in matters of data flow as well as of availability of interfaces within the used software landscape.

8 ACKNOWLEDGMENTS

I would like to thank the following persons & organisations for their contribution: Heinz Studer of RIETER Textile AG for the great input, Jürgen Mewes of Mewes & Partner for lending the software WinMOD and Prof. Roland Siegwart for his financial support.

REFERENCES

[1] J. Gausemeier, J. Lückel: „Entwicklungsumgebungen Mechatronik – Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme“, HNI-Verlagsschriftenreihe, Band 80, Paderborn, Germany, 2000.

[2] K. Henning, K. Kreusch: „Virtuelle Maschinen – Der Werkzeugmaschinenbau erobert die dritte Dimension“, BTU Cottbus, Germany, 1998.

[3] S. Dierssen: “Systemkopplung zur komponenten-orientierten Simulation digitaler Produkte”, Fortschr.-Ber., VDI Reihe 20 Rechnerunterstützte Verfahren, Nr. 358, VDI-Verlag, Düsseldorf, Germany, 2002.

[4] D. A. Bradley, D. Dawson, “Information based strategies in the design of mechatronic systems“, Department of Engineering, School of Engineering, Computing and Mathematical Sciences, University of Lancaster, Bailrigg, Lancaster, UK, 2003

[5] M. Zäh, G. Reinhart: „Mechatronische Produktionssysteme – die virtuelle Werkzeugmaschine“, Seminarberichte iwb, Herbert Utz Verlag, Germany, 2003.

[6] B. Bahr, G. Li, „Virtual Reality systems fort he evaluation of complex mechanical systems“, Proceedings of the 1995 ASME Design Engineering Technical Conference, New York, USA, 1995.

[7] A. Jönsson, J. Wall, G. Broman: „A virtual concept for real-time simulation of machine tool dynamics“, International Journal of Machine Tools & Manufacture 45, 2005.

[8] “VDI 2206: Design methodology for mechatronic systems”, Beuth Verlag GmbH, Berlin, Germany, 2004.

[9] Lückel, Koch, Schmitz: „Mechatronik als integrative Basis für innovative Produkte“, VDITagung: Mechatronik - Mechanisch/Elektrische Antriebstechnik, 29./30. März 2000, Wiesloch, VDI-Verlag, Düsseldorf, Germany, 2000.

[10] Westkämper, Constantinescu, Hummel: “New Paradigm in Manufacturing Engineering: Factory Life Cycle”, Fraunhofer IPA, Chemnitz, Germany, 2006.

[11] “Mewes & Partner”, http://www.winmod.de, Berlin, Germany, last visit January 2007.

[12] “Siemens”, http://www.industry.siemens.de/itps/simulation-center/EN/, Erlangen, Germany, last visit November 2006.

[13] “Rieter Textile Systems”, http://www.rieter.com, Winterthur, Switzerland, last visit January 2007.

Contact: Christian Bacs
Federal Institute of Technology, Institute of Robotics and Intelligent Systems, Autonomous System Lab (ASL)
Tannenstrasse 3
8092 Zürich
Switzerland
Phone: +41 44 632 59 32
Fax: +41 44 632 11 81
e-mail: bacs@mavt.ethz.ch
URL: http://www.asl.ethz.ch