INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

FUNCTIONAL MODELLING FOR EFFICIENT GENERATION OF MECHATRONIC DESIGN AND VALIDATION MODELS OF AUTOMATED PRODUCTION INSTALLATIONS

R. Zafirov, J. Kiefer and M. Eigner

Keywords: mechatronic design, cross-domain engineering, mechatronic validation, digital factory, functional modelling

1. Background and Motivation

1.1 Automated production installations today

Automated production in the automotive industry is nowadays increasingly dynamic. New product variants and product lines are being integrated in current production processes in order to more effectively utilize existing manufacturing equipment. In this way complexity of production processes is constantly growing. This will remain a constant trend, considering the Strategic Research Agenda of Manufuture Germany [Manufuture-D 2007] where adaptive production is stated among german industry's top five research priorities for the 7th EU framework programme.

On the other hand, diminishing product model cycles heavily influence engineering of production systems by setting stricter time limits: with every new product model new production planning is conducted.

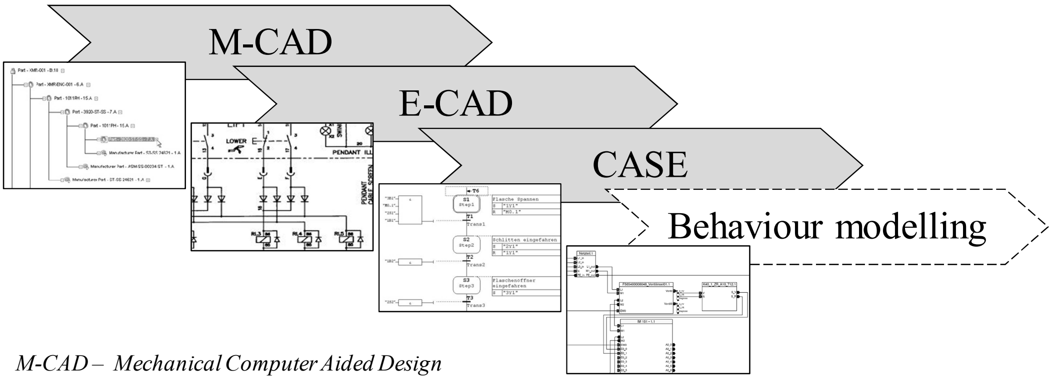
As a result production installations are now engineered anew or refitted to handle new tasks more and more frequently. At the same time, the commissioning stage is impeded by sophisticated production processes as a growing error source for controller programming. The demand for speedy design and early digital validation of automated production installations is therefore very high today.

1.2 Design of automated production installations

All the digital models that are outputs of mechanical engineering, electrical engineering and software engineering of automated production installations are structured differently (Figure 1) and involve different components of a whole installation due to domain-specific levels of abstraction.

Models in M-CAD are generally hierarchically structured in assemblies. Out of the whole installation only components with relevance to mechanical construction, like newly designed (tool or work piece rack, robot gripper) or catalogue components (pneumatic cylinder, robot) are part of the M-CAD model.

On the other hand, E-CAD models contain information on all active components in a mechatronic system: all sensors, actuators and controllers. Electrical design in the area of automation involves selection of suitable components, as well as laying out their power supply and communication interface. Moreover modelling and documenting pneumatics with E-CAD tools is nowadays state of the art. Structures of E-CAD models have a network character, same as the electrical schematics that derive from them.



M-CAD – Mechanical Computer Aided Design
E-CAD – Electrical Computer Aided Design
CASE – Computer Aided Software Engineering

Figure 1. Different models in domain-specific engineering

In the design of automated production installation software is in most cases never modelled in advance but directly programmed onto the PLC (Programmable Logic Controller) with specific programming tools, following international standards [IEC61131-3 2003]. Graphical PLC programming languages (especially the Function block diagram) allow for intuitive, component based programming. Therefore PLC Software has a network structure, built out of interconnected software blocks, each one related to a certain component.

In this sense, E-CAD models and PLC Software are similarly structured and there are certain E-CAD tools in operation, which allow for extensive exchange and even automated generation of hardware configurations and software for PLCs [EPLAN 2009].

Behaviour models are not entirely state of the art. They are being deployed in places for the purpose of design validation. Such models contain algorithms describing all possible states of a component and the actions it performs to change its states when signalled by the controller. In other words, these models incorporate the behavioural capabilities or the uncontrolled behaviour of active components (sensors, actuators) in an installation. With deployment of such models PLC software can be programmed and tested in a digital environment where PLCs interact with behaviour models of the rest of the installation long before it is built.

Similar to E-CAD models, behaviour models involve active components of an installation. Mostly, only components in direct interaction with the PLC (valves, some circuit breakers) and components with relevance to (geometrical) process visualization (hydraulic/pneumatic clamps, cylinders) are included in the behaviour model of an installation. Also similar to E-CAD models, as well as PLC software, the model structure has a network character.

1.3 Motivation

Current domain-specific design models show some deficiencies when it comes to transferring information to one another or bringing together the information they contain.

Meanwhile hardware configuration and software programming for PLCs (CASE) are being very well integrated with electrical design. M-CAD models on the other hand are differently structured than E-CAD models and mostly describe different components. However, there are common components, different aspects of which are described in both models. M-CAD describes 3D geometry and kinematics and E-CAD – the power supply and communication interface within the installation. At the same time, representations of such common components in mechanical and electrical cannot be easily cross referenced between models due to different model structure and domain-specific instantiation techniques.

Furthermore, behaviour modelling is still separated from classic engineering phases although the model structure is similar to E-CAD's. Commonalities with PLC software extend the scope of similar structures, since modelling languages (e.g. Petri nets) and even some of the programming languages (e.g. Sequential function charts) used for behaviour modelling are also common (and standardized) for PLC programming. Despite these facts, behaviour modelling is still being done manually for the most

part, requiring additional effort to conduct thus slowing down engineering and diminishing the added value of digital validation itself.
For a mechatronic validation all mentioned different representations must be working together: mechanical (visualization), electrical (communication), behaviour models and software should interact with each other. Bringing this information together without additional effort may only be possible if relations between all the domain-specific models have been predefined in the engineering phase.

2. Goals of the approach

The approach introduced in this paper aims at meeting today's demands to conduct engineering of automated production installations within stricter time limits. The main goal hereby is to avoid redundant operations throughout the engineering phases and errors during the commissioning phase through efficient generation of digital design and validation models.
The goal is subdivided in two goals. The first one focuses on creating the prerequisites for a smooth digital validation in the foregoing phases of domain-specific engineering. The second one focuses on the generation of additional models, needed for the purpose of mechatronic validation.

Goal 1: Attaining a correlation between the domain-specific digital design models

Seeing how differently domain specific models are built (hierarchical vs. network structures, different aspects of modelling), unifying them would impact the design process within the domains. The goal here is to centrally administer all the information that is relevant for interdisciplinary design without changing domain specific design methods or models.

Goal 2: Further use of the domain-specific digital models for validation purposes

Information about all used components in an installation and how they are related to each other is contained in the digital models as results of the engineering phase. To achieve an efficient generation of models for mechatronic validation (specifically behaviour models), all existent information has to be adopted from the domain-specific models and not recreated manually.

3. Approach introduction

This approach proposes the use of the method for functional modelling not only in the concept phase but also throughout the engineering phase of automated production installations' lifecycle.
Starting off as an abstract, domain-independent description of the mechatronic system [Pahl and Beitz 2002], its primary task should still be the support for finding suitable solutions in design.

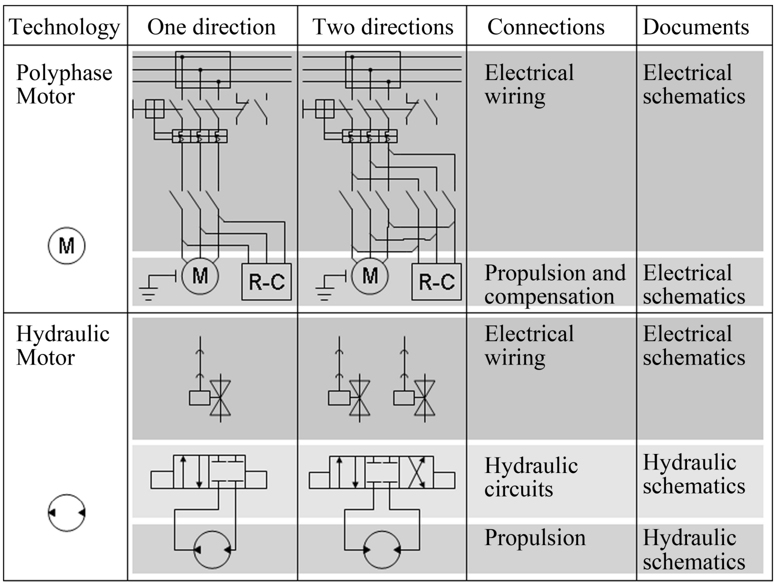


Figure 2. Different technological solutions for the same function [Zäh 2003]

In the particular case of automated production installations this usually represents picking the most suitable technology for a certain function of the installation (Figure 2: electric or hydraulic propulsion). State of the art research in the area of functional modelling concentrates on defining formalized methods for function description and finding ways to administrate product design knowledge in suitable repositories to accelerate, diversify and/or standardize decision finding in product design [Hirtz 2002].
Beyond that, the function structure should be further used to document results as engineering progresses. This task of the function structure should be to store and communicate relevant data between different domains and iteration loops in the engineering phase on the component level (Figure 3).

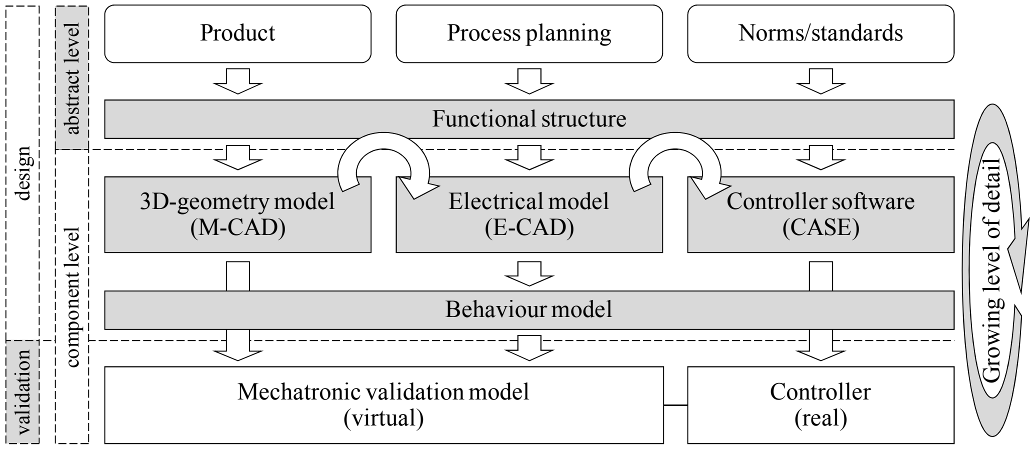


Figure 3. Function modelling for efficient generation of mechatronic simulation models

Thus the function structure can serve as a communication medium between all domains involved in later engineering phases. In this way it can relieve design of redundant tasks related to components that are common for mechanical and electrical digital models. Whenever certain components (sensors, actuators) are instantiated in a domain-specific digital model, this should be documented in the function structure and the information made available for other engineering domains.
Thus, all the data needed for an overall (mechatronic) validation of the engineering results would be cross referenced and easier to reach. This would allow for a more efficient conduct of mechatronic validation. With the gradual increase of information detail throughout the engineering phase, the level of these tests should reach the virtual commissioning of production installations.
Implementing function modelling in the engineering phase to achieve efficient mechatronic simulation model generation is described here in three steps:
Step 1: Hierarchical function structure
According to this approach, a function structure of the automated production installation is built at the beginning of the concept phase. Information about the developed product and results of the manufacturing process planning are used hereby as premises/input information (Figure 3).
Thus, the function structure serves as a common, domain-independent description of the installation and is the foundation for mechanical, electrical and software engineering later on. Each of the resulting domain-specific digital models would then share the same basic structure.
Step 2: Behaviour Model
As already stated, for a mechatronic validation, behaviour models of function-related components have to be put together and cross linked to a network structure, very much like the electrical model of an installation itself. In this step, a behavioural model of the installation is to be automatically generated using the electrical design model to cross link pre-defined behavioural models of single components.
Step 3: Mechatronic validation model
In this step the 3D-Geometry model is further used alongside the Behaviour model for visualisation purposes. Also the controller software (running on a hardware or emulated PLC) can be tested against

the Behaviour model utilizing all input and output signals of sensors and actuators in a production installation (Figure 4).

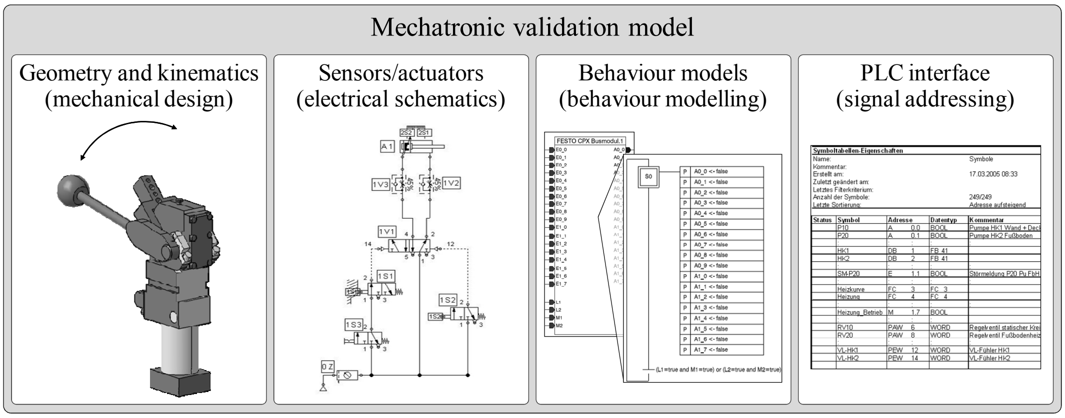


Figure 4. Basic information contents of the Mechatronic validation model

This digital validation model allows (depending on the current stage of the engineering phase) for the validation of geometrical design in terms of collision analysis, as well as the validation of electrical/software design [VDI 2007] in terms of behavioural simulations featuring visualization of motion sequences.

4. Implementation with a practical example

The automated production installation used for this practical example is a robotic welding cell from the automotive body shop. This example covers the benefit of the function structure as a communication medium between mechanical and electrical design.

Figure 5. Overview of the robotic cell that was taken as example

This cell has been designed to function along with an automated guided vehicle system which delivers all five metal sheets placed on a work piece rack. It was useful to give every single piece a shorter name, to easier describe the functions of the cell later on: two engine tunnel pieces (“et 1”, “et 2”), two frame pieces (“fr 1”, “fr 2”) and one base piece are welded together here.
The first part of the example focuses on the whole process in the cell and shows how the phase of process planning contributes to further specifying the function structure. As mentioned in chapter 3, the function structure is constantly evolving throughout the engineering phases. The state of the function structure showed in Figure 6 is intentionally derived from the original process plan documentation for this robotic cell, in order to show how certain tasks in domain-specific engineering gradually turn the abstract character of the structure into a component-based one.

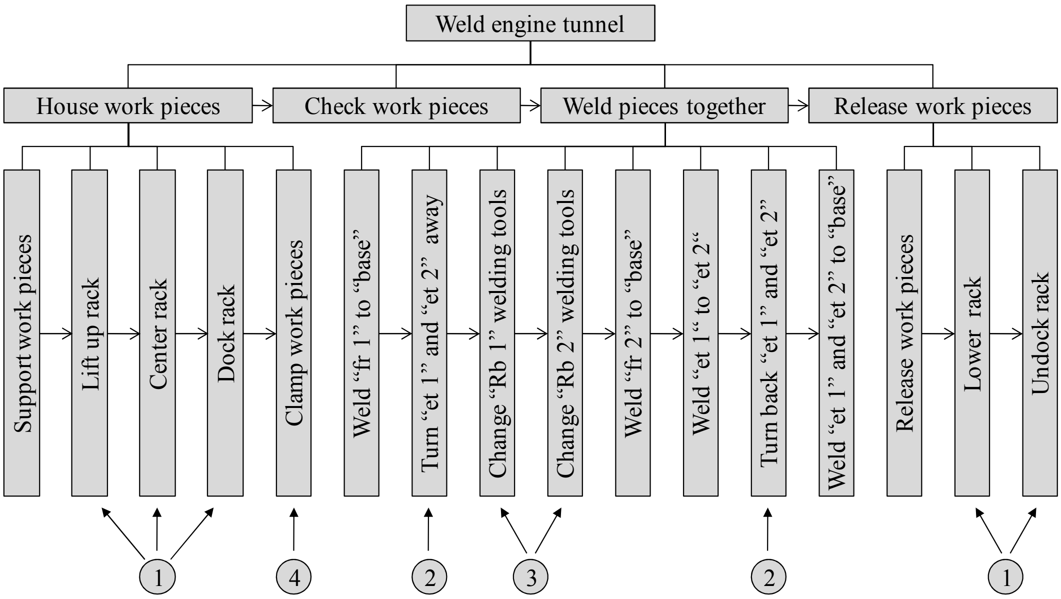


Figure 6. An example of the constantly changing functional structure (numbered functions are mentioned explicitly in the text)

The first hierarchical level shows the main functions of the cell and the second – their sub-functions. The environment of the cell dictates that it has to work together with an automated guided vehicle system. Therefore, a mobile rack needs to be designed, which should be able to dock into the cell’s power supply and controller circuits. So, the rack comes into the functional description of the cell as the first (generically) specified cell sub-assembly. All functions related to the rack (Figure 6, “1”) are considered here as sub-functions of housing and releasing the work pieces.

Throughout the phase of process planning analysis has shown that available welding tools have no access to certain work pieces and therefore both engine tunnel parts should be turned away at 90 degrees, while still clamped together, in order to make room for the robots (Figure 6 “2”). Such decisions directly influence mechanical design. Process planning also dictates how many robots are needed to efficiently conduct the welding job and in which sequence welding should take place. A further example of how the function structure becomes less abstract and more component-related after the phase of process planning is the change of welding tools (Figure 6, “3”) on two of the four robots (“Rb 1, 2”).

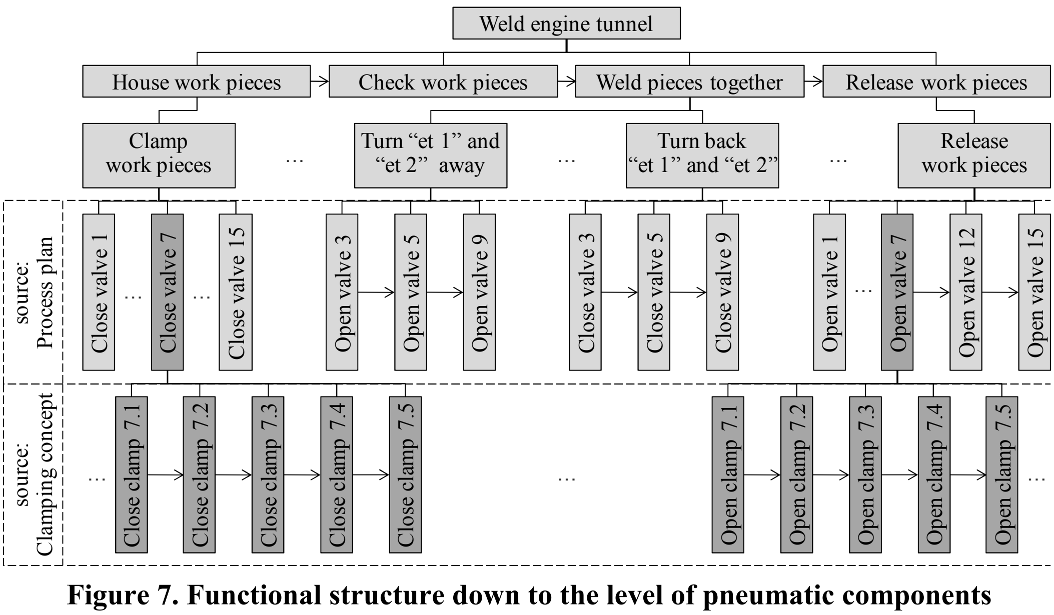


Figure 7. Functional structure down to the level of pneumatic components

The second part of the example shows how a function can support communication between mechanical and electrical design on the component level. Only those of the above listed sub-functions, which are relevant to the use of pneumatics, are shown on Figure 7. “Clamp work pieces” for example, is an abstract function definition from an early phase of the cell’s design. Finding the solution for this problem takes place within the area of mechanical design. In the field of metal sheet welding, the so called “clamping concept” has to be defined. The results of this phase are the exact positions for clamping of the metal sheets, so that they are held stable during the welding process. To clamp the “base” and “frame 1” for example, five pneumatic clamps are needed.
During Process planning it is decided which pneumatic clamps or cylinders should open/close together which leads to determining the number of valves needed. Based on the working sequence, clamps/cylinders are assigned to certain valves in mechanical. Breaking down the functions to component level stores relevant information for electrical engineering, software engineering and behaviour modelling. In this case, information on pneumatic wiring is stored within the function structure (Figure 7, valve 7 with corresponding clamps shown in dark grey).

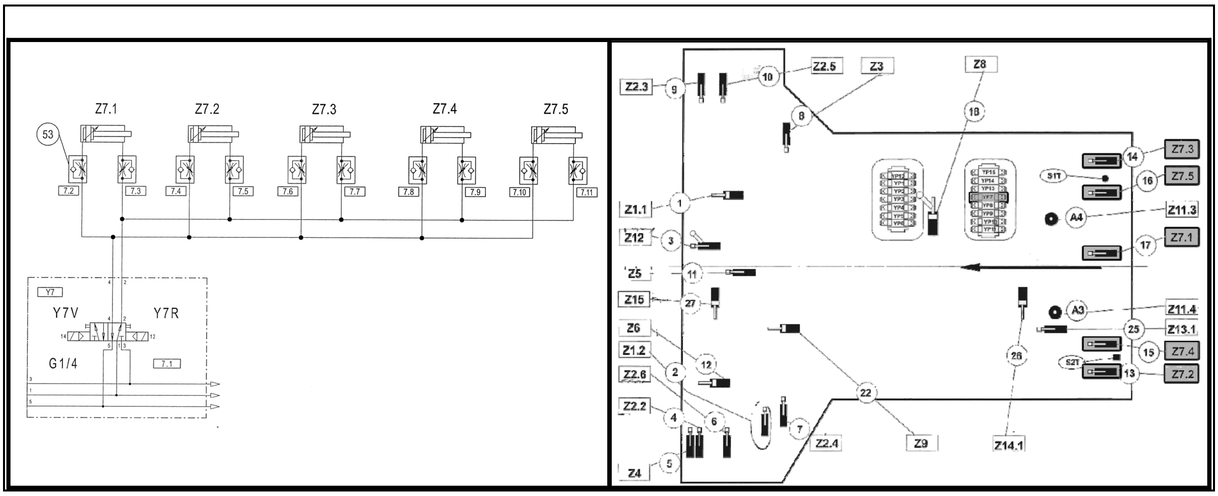


Figure 8. Schematics and rack layout of involved clamps with associated valve in electrical documentation (marked in grey to the right)

The pneumatic circuits shown on Figure 8 (to the left) can be derived from the function structure. Putting the Behaviour model together out of single components’ Behaviour models is also possible.

5. Summary and outlook

Shorter product engineering cycles define stricter time limits for (re-)design and commissioning of automated production installations. Therefore, the demand for efficient digital validation of engineering results from production installations’ design is constantly rising.
After analysing differences and commonalities between different domain-specific digital models of production installations, the approach of domain-independent function modelling as communication platform throughout the engineering process was proposed. An example of a function structure was created using a real automated robotic welding cell from the automotive body shop as example. It was shown how function modelling describes abstract functionalities in the concept design phases of the installation and later on relates to certain components chosen in different engineering domains. On the basis of the simple clamping function it was shown how function structures can store information relevant to more than one domain: valves and pneumatic clamps/cylinders picked throughout mechanical engineering could be indirectly described in the function structure. This example shows some issues within the current approach that need to be tackled in further research work:

- Function models describe processes while domain-specific digital models describe objects (the production installation itself). The example shows clearly how certain components appear more than once in the function structure, related to different functions (Figure 7, “open” and

“close”). A way to derive a modularized structure of the installation based on the function structure would be a possible solution.
- The function structure is meant to be a data storage platform throughout engineering. The structures shown here only contained function hierarchy and sequence. The next step would be to analyze what additional information would be useful to communicate between domains and how to usefully store it under the function structure nodes as metadata.
- A central issue is the digital representation of the function structure. Along with the question “what information to store” within the function structure, the question “how to meaningfully store it” is vital to achieving the goal of efficient model generation for digital validation.

References

EPLAN Software & Service, "Shorter processing times at the SPS", EPLAN Software & Service at the SPS/IPC/DRIVES Fair 2009.

Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., Wood, K.L., "A functional basis for engineering design: Reconciling and evolving previous efforts", Research in Engineering Design 13, Springer-Verlag 2002.

International Electrotechnical Commission, "Programmable controllers - Part 3: Programming languages", ISBN 2-8318-6653-7, 2003.

MANUFUTURE-DE, "Strategic Research Agenda, Roadmap-Prozess Januar 2006 bis September 2007", Management Summary 2007.

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., "Aufstellen von Funktionsstrukturen", Konstruktionslehre, chapter 6.3., Springer Berlin Heidelberg New York, 2002.

Verein Deutscher Ingenieure, "VDI-Richtlinie, Simulation von Logistik -, Materialfluss und Produktionssystemen - Maschinennahe Simulation", Beuth Verlag, Düsseldorf, 2007.

Zäh, M., -F., Grätz, F., Rashidy, H., "An Approach To Simultaneous Development in Machine Tool Industry", International Workshop on Modelling and Applied Simulation 2003, Bergeggi, Italy, 2003, pp. 128-133.

Dipl.-Ing. Radoslav Zafirov
Research Assistant VPE
University of Kaiserslautern, Institute for Virtual Product Engineering (VPE)
Gottlieb-Daimler Straße, Building 44, 67663 Kaiserslautern, Germany
Telephone: +49 (0)631 - 205 3787
Telefax: +49 (0)631 - 205 3872
Email: radoslav.zafirov@mv.uni-kl.de