INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DOMAIN-SPANNING DESIGN TOOLS FOR HETEROGENEOUS SYSTEMS

Torsten Brix, Michael Reeßing
Ilmenau University of Technology (Germany)

ABSTRACT

In the past it was often adequate to assemble an overall system from separately developed and optimized parts. However, recent developments in engineering show the need to integrate mechanical, optical, electrical, electronical and software components. This new quality of interdisciplinary collaboration requires new computer-aided, phase- and domain-spanning tools for modeling, analysis, simulation and optimization of complex design objects particularly in the early phases of product development. The article covers the first intermediate results of the ongoing research concerning a phase-overlapping design software tool for heterogeneous systems that supports the transition from the solution principle to embodiment design. Finally more general remarks, theses and questions concerning procedures, methods and tools for a domain-spanning and phase-overlapping design are presented to encourage a discussion. These thoughts base mostly on common statements and views in german literature.

Keywords: Heterogeneous Systems, Early Phases of Product Development, Constraint-based Modeling

1 INTRODUCTION

For computer-aided modeling and calculation of heterogeneous systems different approaches are known. Some of them already allow a holistic development process for (usually mechatronic) components and products.
The following categories exist:

- general simulation systems,
- specialized simulation systems,
- parallel application of multiple simulation systems,
- guidance and assistance systems and
- integrated solutions.

For the design and particularly the simulation in early phases general simulation systems can be applied (e.g. MATLAB, COMSOL, Modelica). These systems are flexible since modeling is done by entering source/script code, equations or schematic diagrams that describe and simulate heterogeneous systems on a functional level (Figure 1Figure 1). However, this approach leads to a number of disadvantages. It is often difficult to derive equations that represent functional properties and to incorporate structural properties (e.g. function-relevant positional relations of mechanical and optical components). Also, there is usually no support for the transition from the functional description to the embodiment.
Specialized simulation systems (e.g. SPICE, ALASKA, DIFFRACT, Working Model) are usually not suitable for holistic modeling and simulation of heterogeneous systems. For instance, the modeling system for electronic circuits SPICE allows limited handling of non-electronic components that have to be transformed into equivalent electronic or electric circuits. It is not possible to model components for feedback control systems. Serious limitations concerning a uniform equivalent description of heterogeneous systems also exist in dynamic simulation systems (e.g. ADAMS, ALASKA, SIMPACK, ITI-Sim), in software for control theory (e.g. FSIMUL, SIMULINK) or in optics simulation systems (e.g. OSLO, ZEMAX, DIFFRACT). A few interdisciplinary tools exist for modeling and simulation in mechatronics. One example is CAMel-View [12] that allows connecting embodiment design to real-time simulation. However, most of these systems show significant

drawbacks and problems concerning a user-friendly domain-specific model representation or support of the transition between the solution principle and the embodiment.

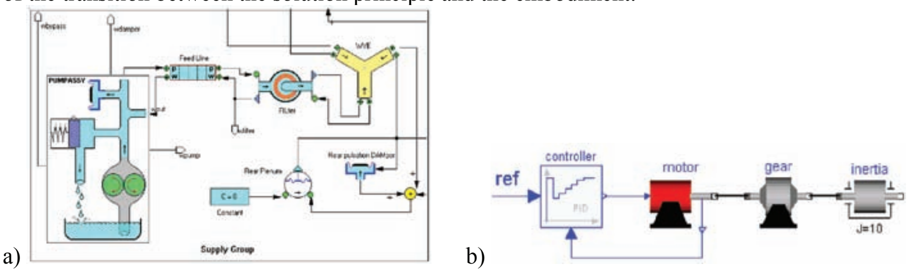

Figure 1: General simulation systems
a) Example model of a fuel injection system in ACSL [1]
b) Modelica model of a simple drive train [2]

The parallel application of multiple simulation systems considers the necessary multidisciplinarity of the design of heterogeneous systems as the user applies a number of different, domain specific software that support modeling, analysis/simulation and evaluation/optimization. This approach demands a high degree of design experience and the mastery of many software products. Also, redundancies and loss of information are inevitable in such a process. To avoid the resulting additional expenses, it is reasonable to join the different software products that cover only narrow fields of representation, calculation, simulation and optimization.

One possible approach for this is to employ guidance and assistance systems [3]. These provide a methodical guide with specifications for merging the data of a design object in a uniform repository of data. The core of such systems is usually a CAD system that is linked with other software products to support the design of certain products or product groups.

Around the CAD software world many powerful tools emerged that usually support the later phases of the design process. One of the important developments in recent years is the Digital Mock-Up/Functional Mock-Up. Such integrated solutions that are usually add-on modules for CAD systems allow multi-body simulation, finite element method, the design of drives and controllers as well as examinations of usability and assembly processes. Key technologies in the area of real-time applications are Virtual and Augmented Reality. Virtual Engineering not only features visual but also acoustic immersion using techniques like Wave field synthesis combined with haptic, visual and acoustic interaction methods. The increasing importance of saving, providing and applying of product and process knowledge is pointed out by keywords like Knowledge-based Engineering, Product Data Management, Product Lifecycle Management and Product Lifecycle Engineering [4].

The listed approaches can be seen as intermediate steps towards the aim of a continuous computer-aided product development process with a minimum of iteration cycles. Innovative trends like developments in the area of self-optimizing systems or the application of novel, “intelligent” materials that demand a holistic and therefore domain-spanning consideration of product properties define new requirements for future tools in computer-aided product development.

A major contribution came from the Collaborative Research Centre SFB 241 “New Integrated Mechanic–Electronic Solutions for Mechanical Engineering Systems” [13] that aimed at the integration of electronic and mechanical components using complex information processing.

However, despite enormous advances [e.g. 5, 6, 7, 8, and 9] the mentioned approaches show deficiencies concerning the domain-overlapping description of heterogeneous systems and the phase-overlapping modeling from the functional description to the embodiment design.

2 HOLISTIC DESIGN OF HETEROGENEOUS MOTION SYSTEMS IN EARLY DESIGN PHASES

There are many possible ways for the holistic handling of heterogeneous systems. This section presents three different approaches that are subject to research activities at the Engineering Design Group of the Ilmenau University of Technology. The concepts are focused on the solution principle as an idealized representation (using graphic symbols) of the system structure that describes function relevant elements and couplings.

2.1 Integrated approach with homogeneous model representation and simulation environment

A prerequisite for holistic modeling of technical systems is a model representation that covers all relevant aspects of the involved domains. On the abstraction level of the solution principle it is purposeful to utilize structure-oriented modeling. The first presented approach uses a model representation that derives from the requirements of certain calculation methods or simulators. An example is the design tool MASP (Modeling and Analysis of Solution Principles, [11]), that was initially created to design motion systems and is based on a geometric constraint solver. Components from other domains can be implemented in this software if it is possible to formulate a geometric substitution model. For instance, MASP handles optical components using laws of geometric optics.

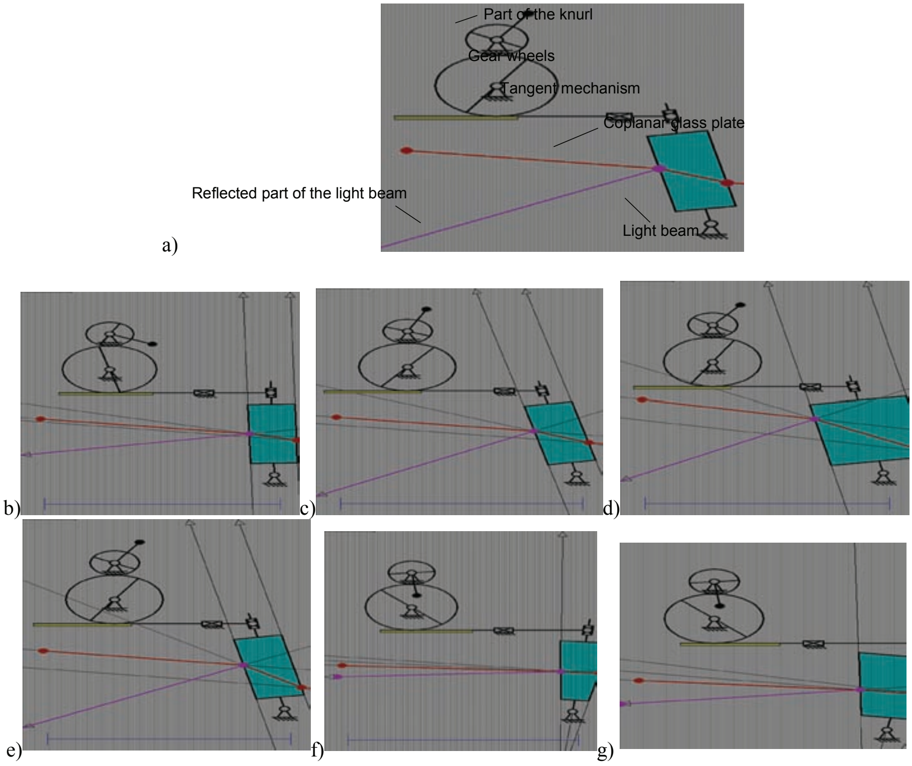

Figure 2: Modifications of the solution principle of a micrometer modeled in the design system MASP (the shown geometric sub-elements like directed lines illustrate the underlying constraint model).

***a)** and **b)** Calculation of different positions of the micrometer by interactive rotation of the knurl.*
***c)** The variation of the parallel offset with immovable adjustment mechanism and constant refraction index causes a change of the glass thickness.*
***d)** The variation of the parallel offset with immovable adjustment mechanism and constant glass thickness causes a change of the refraction index.*
***e)** The coplanar glass plate changed into an optical wedge.*
***f)** and **g)** The variation of the parallel offset with immovable adjustment mechanism and constant refraction index causes a change of the wedge thickness*

Creating such constraint-based models from geometric base elements does not match the way most designers think and should therefore not be their task. As a solution, the concept introduces two specialized views: a user-oriented and a calculation-oriented representation with an automatic transformation between both (Figure 2). The user-oriented view consists of graphical symbols for solution elements from the involved domains. The (calculation-oriented) constraint representation behind these symbols remains invisible for the user.
The result is an interactive set of build blocks with integrated simulation capabilities that supports a simple and fast conceptual design of solution principles. Such an integrated design tool is limited by the features of the used solver and the possibility to formulate suitable substitution models. Extending the tool to handle other domains can be difficult.

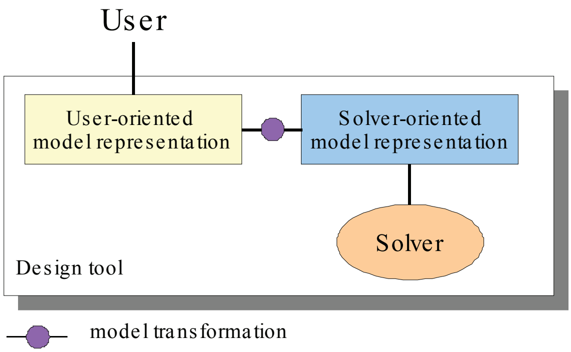

Figure 3: Structure of an integrated, solver-specific design tool

2.2 Platform with neutral model representation and interfaces for calculation modules

Another architecture for a design tool that has been implemented as a prototype is an extension of the concept from section 2.1. It adopts approved concepts like the building blocks system of predefined solution elements and the level of abstraction (solution principle).
For a more flexible incorporation of different domains, the concept splits model information into a user-oriented view, a calculation-oriented view and a neutral model representation that is independent of the requirements (e.g. data structures) of a particular calculation method or constraint solver. It is hardly possible to find a solver to evaluate this model as a whole. Therefore, task-specific calculation modules handle parts of it. A conversion layer interprets the semantics, parameters and relations of the neutral model elements and transforms the neutral model description into a solver-specific representation.
The neutral model representation is also a synchronization repository for the data stored in the calculation modules so that it is only necessary to implement a limited number of converters. Another advantage is that the model description core as well as the visualization and conversion layers can be extended using plug-in interfaces. As a result, it is possible to include additional solution elements, calculation methods (even from other domains) into the design tool at any time.

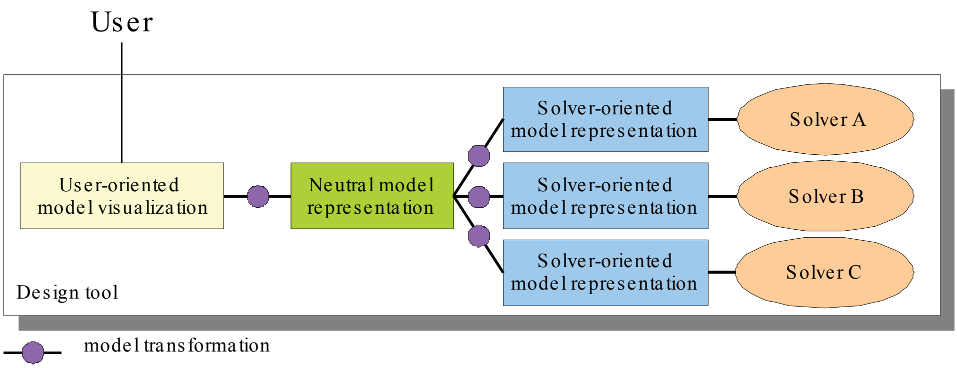

Figure 4: Design tool as a platform for calculation modules

While extending the neutral model representation is simple in most cases, adding new calculation methods or solvers requires them to be available as program libraries, to have communication interfaces or to be implemented by the plug-in author. Also, it is necessary to coordinate the calculation modules.

2.3 Central entity for exchange and coordination

A different proposal extends the concept from section 2.2 to be a coordination center for existing simulation tools. The design tool serves for the communication with the designer and hides the involved simulation tools behind a uniform user interface (Figure 5). Such ideas already existed in the past [e.g. 14, 15] and need to be analyzed and adapted regarding new user requirements and today's development stage of hardware and software.

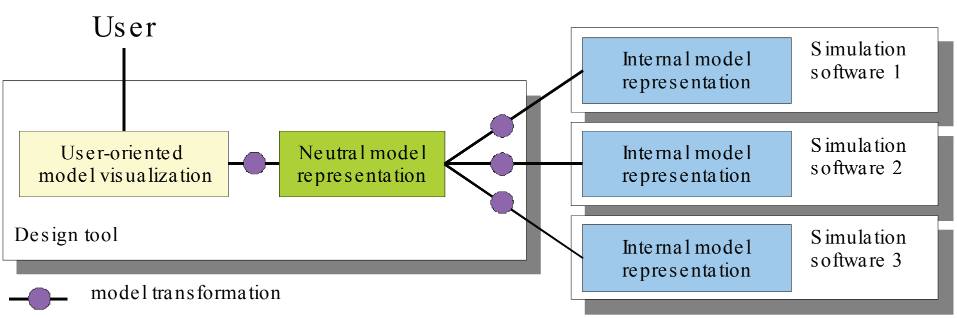

Figure 5: Design tool as a coordinator of existing simulation tools

The synchronisation of the involved tools is performed using an exchange format that is based on the neutral model representation. To avoid loss of information in the conversion process it is necessary that parts of the model which are unknown or irrelevant for a particular tool will not be removed but passed on without changes. A suitable medium could be the Document Object Model of the XML (eXtensible Markup Language), that allows general handling of unknown information.
The advantage of this approach is the combination of a central model and the extensive features of existing software tools without the need to learn the specifics of these (user interface, modeling etc.). On the other hand, such a design tool demands prerequisites from the software producers of the involved software packages that are not easy to fulfill (interfaces to access internals, documentation). Also, conversion of relevant partial models into and from the representations used by the external software is hardly possible without loss of information.

2.4 Neutral model representation

Usually, the internal data structures of the simulation methods employed by a design tool are not a sufficient representation of the complete solution principle. They only store certain task-specific aspects of the solution according to mathematic or algorithmic requirements of the calculation method. To represent all the information of the solution principle, a general structure-oriented concept is more appropriate. This section presents the basic concepts of a possible approach that proved to be suitable in an implemented prototype of a design tool for heterogeneous systems.

The model representation is based on models elements or solution elements of the design tool's building blocks system (section 2.1 and 2.2). They contain semantics, function-relevant base elements and parameters and relations to other model elements (Figure 6). The model of the product is composed as a graph of such solution elements. Typical examples of components represented by solution elements are joint, links, optical mirrors and lenses, sensors, controllers or actuators.

For each domain there is a predefined (but extensible) set of base elements that allows a uniform description of domain-specific components (solution elements). For instance,

- Kinematics: scalar, vector, coordinate system, rigid body,
- Optics: optical media description, boundary surface, optical axis, ray of light.

Solution elements parameterize such base elements and combine them into logical units (parts) that feature semantics familiar to the user (e.g. the part "plano-convex lens" contains a planar and a spherical boundary surface).

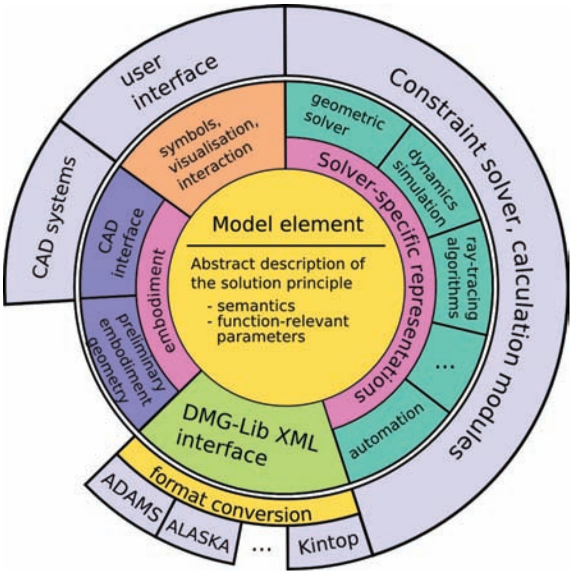

Figure 6: Solution element

Solution elements can be assembled into complex solution elements that may be simple groups or parts with additional semantics and parameters (e.g. "cam pair" composed from a cam and a follower, "Cooke-triplet" of three lenses with combined focal length parameter).

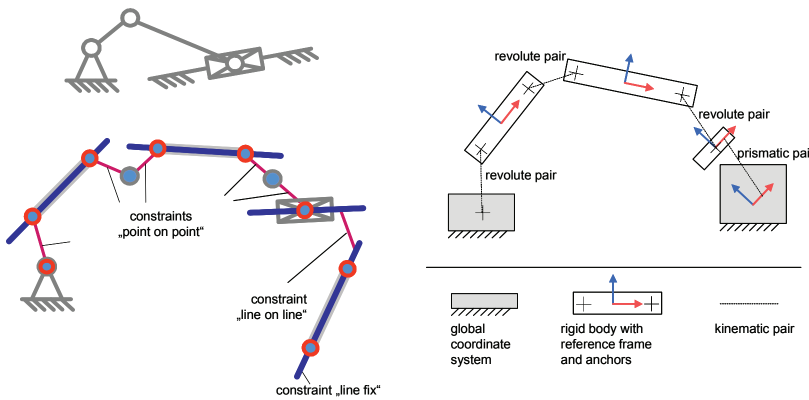

Figure 7: Transformation of the general model representation (upper left corner) into specialized representations for geometric constraint solvers (left) and multi-body simulation (right)

To make solution elements available to solvers and calculation methods the concept introduces extension elements that transform the general model representation of the solution principle into a solver-specific representation (example in Figure 7). These extensions could be considered as "plug-ins" that add new aspects or views to the original solution element, including required additional information (e.g. mass distribution for dynamics simulation). In a similar way it is possible to add extension elements for visualization and interaction, links to other levels of abstraction (e.g. embodiment features) or exporters for external formats.
The result is an extensible domain-spanning representation of the solution principle of heterogeneous systems that may also serve as a basis to estimate the behavior of the system by simulation.

2.5 Integration of calculation modules

To simulate the modeled solution principle of the product it is not only necessary to create solver-specific representations of the neutral model elements, but also to coordinate the involved calculation modules. For coordination there is a set of concepts that needs to be supported by each calculation module. The most import ones are:

- a time concept – to define simulation steps and time spans, chronological mapping of events,
- a consistency concept – a model is in a consistent state,
- the differentiation of changes of topology, of parameterization and of state and
- the description of dependencies.

Not all calculation modules meet these requirements. A possible solution is to supply solver-specific adaptors that add or interpret missing concepts to make calculation modules accessible through a uniform interface (Figure 8). As an example, the geometric constraint solver used in the prototype of the design software does not feature a time concept. It reaches any state of the model within one calculation step. Therefore, it is the task of the adaptor to map geometric changes to a time basis in order to describe values like velocities, accelerations etc.

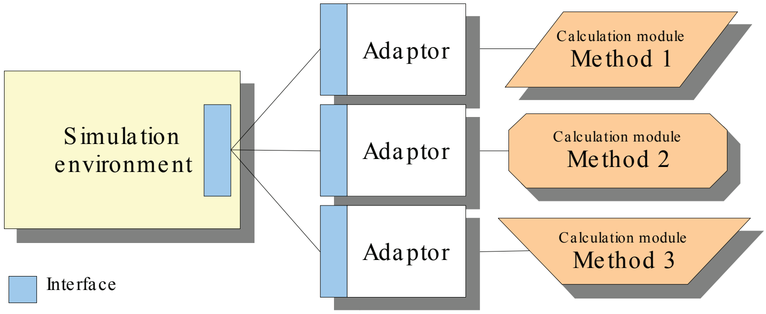

Figure 8: Unification of calculation modules

3 DISCUSSION ON DOMAIN-SPANNING AND PHASE-SPANNING DESIGN

The design of complex heterogeneous products is a very ambitious task and demands advanced skills from designers from several domains. Profound technical knowledge and formed soft skills are important prerequisites for a successful work of designers. In addition the use of suitable design procedures, methods and tools is essential. These support and improve:

- the problem solving by illustration, verification and documentation of mental models,
- the communication between designers and other involved persons,
- the exemption from routine activities,
- the competent handling of domain-spanning complexity by means of a goal-oriented approach and appropriate information processing.

Since many decades there is a discussion about a goal oriented design process and the use of suitable methods and tools to support a domain-spanning and phase-overlapping design. Even in the forties and fifties of the last century there were already books about approaches for a systematic problem-solving in the design process (amongst others Wögerbauer [16], Bischoff/Hansen, Kesselring [18], Matousek) that discussed domain-spanning and phase-overlapping design and emphasized their great importance. The publications primarily base on experiences in the design of mechanical, electro-mechanical and opto-mechanical products. These books were the starting point of newer approaches, which also cover a comprehensive, process-oriented structuring of the whole design process (amongst others Hansen [17], Hubka, Rodenacker, Ehrlenspiel, Koller [19], Pahl/Beitz, Roth). Noticeable is, that the holistic design view is complemented with ideas from the system theory. Background for this was the first experiences in the design of mechatronical, adaptronical and micro-technical products as well as first experiences in the use of computer-based tools, which were driven by the fast developments in electronic/computer technology, cybernetics, materials and manufacturing technology.

Authors of current publications, especially about the domain-spanning design of mechatronic products have difficulties to continue the research based on these newer approaches [8, 13, 20 and other]. Normally, only the authors Pahl/Beitz [21] and the VDI-guideline 2221 "Systematic approach to the development and design of technical systems and products" [22] are listed in the references. A more profound discussion of the extensive literature in design methology is avoided on the ground that for a mechatronic product a different design process is needed than for a conventional mechanical engineering product. Have products of conventional mechanical engineering like a steam turbine (requires collaborative work of experts in fluid mechanics, material science etc.) or a microscope (needs a cooperation of physicians, opticians, precision technician etc.) an identical design process? Definitely no, because for each problem or problem class suitable procedures, methods, tools must be selected, combined and adapted. The VDI-Guideline 2206 "Design methodology for mechatronical products" [23] promotes a new design process for mechatronical systems with the statement, that the complexity in comparison to mechanical systems is characterized by a "greater number of coupled elements". Is the calculator of Leibnitz (solved all basic arithmetic operations) from the 17th century not as complex as a controlled catalytic converter used in a cars? (In the eighties of the last century experts needed months to understand the Leibniz machine in detail for a replica.) If the answer is „yes", then why? Furthermore, the VDI-guideline 2206 points out that designing mechatronical products requires the collaboration of experts from different scientific disciplines. But this applies also to almost all areas of the “classical” mechanical engineering, even though the emphasis is not always placed on digital information processing or electronic.

The pressure to establish a border between the “classical” design methodology and the design methodology for mechatronical products leads to meaningless names for design approaches (V-model, Y-model, spiral model, 3-level-modell etc.). Herein lies a certain symbolic significance for the present state of the research in design methodology to support the design of heterogeneous products.

A reason for these developments gives the “classical” design methodology by itself, because it provides not enough or no remarks, assistance etc. for the massive and meanwhile essential use of computers in all domains. Many publications, which contain comparative views on different approaches for a domain-spanning design, focus normally on specific differences between the approaches [as example 8, 13].

The used terminology, the boundaries of work steps and phases as well as the using of efficient methods and tools are analyzed as characteristics for the differentiation of approaches. Such a view often complicates the identification of the essence of an approach or methodology, which is affected by specific assumptions and experiences. Therefore it would be better to elaborate the similarities between approaches as well as to collect the advantages and disadvantages of several approaches during the handling of certain problems or design tasks. Finally it is always a matter of the forming of the problem-solving-cycle adapted to the task and further constraints.

Perhaps the previous design approaches could be understood as adapted views of the same general design approach. The investigation of this assumption is an extremely ambitious task, because an independent analysis and evaluation of findings and doctrines is required. The necessity of the adaptation of design approaches to the concrete design problem as well as to the skills of the designer or the design team is not sufficiently discussed. This often leads to distorted evaluations of design approaches/methodologies. For instance the persistent opinion still exists, that the approach in the VDI-guideline 2221 forces a pure sequential work from one design state to the next as a stringent sequence and that the guideline is only suitable for the “classical” mechanical engineering. The general approach in the VDI-guideline 2221 is only a skeletal structure. Based on this a specific, problem adapted, chronological and logical approach must be developed. Only a holistic analysis of the design task concretizes the design approach including needed methods and tools. Furthermore, the concretized approach must be adaptable for modified or new constraints or conditions. So there can not be only one design methodology for mechatronic products, which satisfies all requirements of the product design in a range from a simple digital clock to a global positioning system.

The existing deficiencies in the design process (in practice) are often not a problem of the design method, but rather a problem of the work organisation (e.g. project management), of used tools (e.g. problems of data transfer between software programs, domain specific software). These circumstances do not free the “classical” design methodology from deficiencies like providing insufficient information about the optimal selection and adaptation of procedures, methods and tools for certain problems. Would significant information exist about this then the topic “mechanics versus mechatronics” could be considered to be such an application or adaption scenario.

Especially the use of software tools is connected with many open questions including the insufficient integration of tools for a phase-overlapping and domain-spanning design. Other open questions are for example:

Product information

- Which information about the product does the developer really need and at what time? How to prevent disorientation of the designer that can arise from information overflow or from misleading information? How to avoid loss of information in conversion processes?

Model representation, visualization and interaction

- Which model representations for phase-overlapping design are suitable from the designer’s point of view? What are the representations for the design of heterogeneous systems? Which of them are rather solver-specific (e.g. FEM model)? Is it possible to clearly distinguish between model representation and PDM?
- Which model views are really necessary? How to adequately visualize the model?
- What is a suitable way for product model parameterization? How should the user access it?
- Which heuristics and rules may be involved to support the user’s interaction with the model (separation of model representation, calculation and visualization leads to ambiguities)? How to input and manipulate faster / easier / more definite?

- Which are the proficiencies and skills a designer of heterogeneous products really needs? What knowledge does he need about the mathematical methods (FEM, MBS, constrain solver etc.) employed in a design tool?

Calculation modules (solvers) and interfaces

- What calculation methods in which phase of product development are necessary to determine product characteristics? How to access solvers or calculation modules?
- Which interfaces are necessary? Is it possible to establish new ideas for standards?

Answering these and other questions requires thinking outside the (domain-specific) box and detaching from the limitations and deficiencies of present software techniques and the different more or less methodic approaches. Only then it becomes possible to precisely specify the general and problem-specific requirements for an ideal design tool supporting the whole process of product development. This could at least help to determine targets for further developments in this area, regardless if they can be reached in the near future.

4 CONCLUSION

The article presents concepts as well as a prototype implementation of a software tool that primarily aims to increase the effectiveness of the design of multi-domain solution principles and therefore also improves embodiment design of heterogeneous product. The concept focuses on the integration of modeling operations and calculation methods for a holistic system description of design objects to avoid disadvantages of existing approaches (e.g. loss of information). The model representation is based on a constraint-based structural description that holds the properties of heterogeneous models as a whole.

The starting point for the development of such and other design tools are in many cases considerations about methodical procedures. In this area exists a number of unclarities and open questions. Some of them are mentioned to point out that there is demand for in-depth research pursuing the long term goal of a user-friendly, computer-aided, phase- and domain-spanning design tool.

REFERENCES

[1] ACSL Graphic Modeller User's Guide, 2007 (AEgis Technologies Group, Inc., Huntsville).

[2] *Modelica and the Modelica Association*. http://www.modelica.org/index.shtml, 2007.

[3] G. Pahl and W. Beitz: *Konstruktionslehre*, 2003 (Springer, Berlin).

[4] Franke, H.-J.; Gausemeier, J.; Krause, F.-L.: *Innovationspotenziale in der Produktentwicklung*, 2006 (Hanser-Verlag, München).

[5] Koch, T. *Integration von Konstruktion und mechatronischer Komposition während des Entwurfs mechatronischer Systeme am Beispiel eines integrierten Radmoduls*. Ph.D. Thesis, University Paderbor, 2006 (HNI, Paderborn).

[6] Hestermeyer, T. *Strukturierte Entwicklung der Informationsverarbeitung für die aktive Federung eines Schienenfahrzeugs*. Ph.D. Thesis, University Paderborn, 2006 (HNI, Paderborn).

[7] Schlautmann, P. *Entwicklung eines neuartigen dreidimensionalen aktiven Federungssystems für ein Schienenfahrzeug*. Ph.D. Thesis, University Paderborn, 2006 (HNI, Paderborn).

[8] Möhringer, S. *Entwicklungsmethodik für mechatronische Systeme*. Habilitation, University Paderborn, Volume 156, 2005 (HNI-Verlagsschriftenreihe, Paderborn).

[9] Shen, Q. *A Method for Composing Virtual Prototypes of Mechatronic Systems in Virtual Environments*, Ph.D. Thesis, University Paderborn, Volume 193, 2006 (HNI-Verlagsschriftenreihe, Paderborn).

[10] Brix, T.; Döring, U.; Reeßing, M. Multi-Stage Modeling in the Early Design Phases. In 14th *International Conference on Engineering Design* – ICED03, Stockholm, 2003.

[11] Brix, T.: *Feature- und constraint-basierter Entwurf technischer Prinzipe*, Ph. D. Thesis. Ilmenau University of Technology, 2001 (ISLE, Ilmenau).

[12] Fa. Ixtronics: *CAMeL-View*. http://ixtronics.de/cms/website.php?id=/de/index/produkte/camelview.htm, 2008

[13] Isermann, R; Breuer, B.; Hartnagel, H.-L. *Mechatronische Systeme für den Maschinenbau.*, 2002 (WILEY-VCH, Weinheim).

[14] Seifert, H.; Kruse, D.; Wehmöller, M. *Netzwerkbasiertes Kommunikationskonzept zur integrierten Produktentwicklung*. VDI-Z 134, No. 10, October, 1992, pp. 89-96.

[15] Swienczek B., Arnold F., Kilb T. and Janocha A. T. Maßgeschneiderte CAx-Systeme auf Basis von Komponenten. In *2. Workshop Konstruktionstechnik: Innovation - Konstruktion - Berechnung*, 1998 (Shaker, Aachen).
[16] Wögerbauer, H.: *Die Technik des Konstruierens*, 2nd edition, 1943 (R. Oldenbourg, München).
[17] Hansen, F.: *Konstruktionswissenschaft – Grundlagen und Methoden*, 2nd edition, 1974 (VEB Verlag Technik, Berlin).
[18] Kesselring, F.: *Technische Kompositionslehre*, 1954 (Springer, Berlin).
[19] Koller, R.: *Konstruktionslehre für den Maschinenbau*, 1998 (Springer, Berlin).
[20] Gausemeier, J.; Frank, U. Domänenübergreifende Spezifikation der Prinziplösung selbstoptimierender Systeme unter Berücksichtigung der auf das System wirkenden Einflüsse. In *Mechatronik 2005 - Innovative Produktentwicklung*, VDI-Bericht, Volume 1892.1, 2005 (VDI, Düsseldorf).
[21] Pahl, G.; Beitz, W. *Konstruktionslehre: Methoden und Anwendung*. 5. edition, 2003 (Springer, Berlin).
[22] VDI-Gesellschaft Entwicklung-Konstruktion-Vertrieb . *VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*, 1993 (Beuth-Verlag, Berlin).
[23] VDI-Gesellschaft Entwicklung-Konstruktion-Vertrieb . *VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme*, 2004 (Beuth-Verlag, Berlin).

Contact: Torsten Brix or Michael Reeßing
Ilmenau University of Technology
Department of Mechanical Engineering
PF 10 05 65
D-98684 Ilmenau
Germany
Phone: +49 (0) 3677 69 1263
Fax: +49 (0) 3677 69 1259
E-mail Address: torsten.brix@tu-ilmenau.de, michael.reessing@tu-ilmenau.de
URL: http://www.tu-ilmenau.de/konstruktionstechnik/

Torsten is senior engineer of Engineering Design at the Ilmenau University of Technology (IUT). He teaches and researches in several fields of engineering design (e.g. design methods and tools, cost estimation, weak point analysis and error prevention, production oriented design). Michael is computer scientist of the Engineering Design Group at IUT. His main research interest is computer-aided design (Constraint-based design, behavioral design, modeling of heterogeneous systems) to support the early phases of the design process.