

Cross-Domain Dependency Modelling – How to achieve consistent System Models with Tool Support

Rainer Stark, Grisca Beier, Thomas Wöhler
Fraunhofer Institute for Production Systems
and Design Technology
Pascalstr. 8-9, 10587 Berlin, Germany
rainer.stark@ipk.fraunhofer.de

Rainer Stark, Asmus Figge
Technische Universität Berlin
Chair of Industrial Information Technology
Pascalstr. 8-9, 10587 Berlin, Germany
rainer.stark@tu-berlin.de

Copyright © 2010 by Rainer Stark. Published and used by INCOSE with permission.

Embedding and integrating functionalities from different domains has led to a significant increase in information that developers have to deal with. It is hardly possible to oversee implications of activities like change requests since influenced components from other domains are difficult to identify. This calls for an approach that supports product developers with methods and tools helping to analyse, control and primarily build up complex systems and the interdependencies between their comprising models from various domains.

The focus of the presented approach, developed in the project ISYPROM, is the integration and connection of different partial models, used in product development. In doing so, implications of changes become evaluable, potential errors can be identified and information presented to the developer can be reduced. Special emphasis has been put on models describing requirements, functionalities, product structures and especially their interdependencies. Methods and tools focus on usability, information reduction and quality aspects.

Introduction and Motivation

A challenge many companies face today is the increase of complexity. It is defined as a property of a system depending on the amount of system elements, the number of relations between them and the multitude of its possible states (Ulrich and Probst, 1995). Concerning mechatronic product development there are numerous sources for complexity to be found, like the product with its augmented, cross-domain functionalities or the development process itself.

Against this background, embedding and integrating functionalities from different domains has led to a significant increase in information that developers have to deal with. Furthermore it is impossible to oversee the implications of activities like change requests since influenced components from other domains cannot be identified. Thus, changes in these complex systems are very difficult to handle (Maurer 2007).

This problem gets amplified by the fact that product developers need to deal with several specialised tools which lack continuity in software communication. This originates from the fact that originally traditional product development processes have been virtualised without any impact on the modelling procedures, to ease their introduction and support their acceptance by developers. In the course of this virtualisation the number of supporting tools has grown enormously, bearing powerful tools offering special features for one or more sub-processes.

Additionally, departments need to establish collaboration with partners and at the same time increase the level of virtual support along the entire virtual product lifecycle. Frequently this leads to uncontrolled technological and collaborative growth causing an increase of heterogeneous isolated software solutions.

While mechatronic products and development processes become more complex, quality

assurance and risk assessment in virtual product creation have not reached the same maturity level as in manufacturing and assembly yet. This imbalance, which leads to a negative impact on product and process robustness of digital product development as well as a low risk safeguarding, can be attributed to several factors: integrated, parallel, flexible workflows and an increase in variety, causing product complexity and versatility.

The situation described calls for an approach to support product developers with methods and tools helping to analyse, control and primarily build up complex systems and the interdependencies between their comprising models - focusing on the modelling of cross-domain dependencies while avoiding the provision of “just another tool” aiming to replace well established authoring tools.

For this reason the Federal Ministry of Education and Research has set up the framework concept “Research for Tomorrow’s Production” in which the project ISYPROM is integrated (Karlsruher Institut für Technologie 2009). ISYPROM focuses a model driven process and system design to support Systems Engineering and to accelerate innovation. The main target however is the integration and connection of various modelling methods used in industrial reality creating context-oriented views on system models that can be reused in later process phases. The communication between different modelling tools will be supported through Product Lifecycle Management (PLM) using neutral, standardised interfaces when possible. The aim of the project is to develop an approach where information from business processes can be used in product development processes and vice versa. Therefore a linkage between business models, system specifications and models from other disciplines involved is required, to allow reuse of generated information as well as bidirectional navigation between different model types (ISYPROM 2009).

In the course of this project the described deficits of today’s development methods were affirmed by industrial partners. For this reason, the focus of the approach, presented in this paper is the integration and connection of different partial models, used in product development. In doing so, implications of changes become visible and evaluable, potential errors can be identified and information presented to the developer can be reduced. Special emphasis has been put on models describing requirements, functions, product structure and especially their interdependencies. Methods and tools, developed for this purpose, need to focus on usability, information reduction and quality aspects.

They are researched and implemented in cooperation of the Fraunhofer Institute for Production Systems and Design Technology, the Technische Universität Berlin, Chair of Industrial Information Technology and InMediasP GmbH (InMediasP 2009).

State of the art and research

There are several procedure models for the development of mechanical, electronic or software components that are state of the art. Nonetheless their use for the development of mechatronic products is rather limited, since they have been specified by the respective domains. This is due to the fact that used methods are usually not transferable and much needed coordination of domains is not supported.

In mechanical engineering the design process according to Pahl, Beitz, Feldhusen and Grothe (2007) or VDI 2221 (1993) is well-established. This procedure model consists of a sequence of four main phases that are further detailed by design steps. Those deal with the definition of the requirements, the conceptual and the detailed design of the product as well as the preparation of manufacturing and product documentations (Pahl, Beitz, Feldhusen, and Grote 2007), (VDI-Gesellschaft Entwicklung Konstruktion Vertrieb 1993).

The “Münchner Vorgehensmodell” (MVM) is another procedure model consisting of seven elements. Each step represents a different design step of the development process. In contrary to Pahl and Beitz a net-like illustration is used to explicitly emphasise that

development processes are not characterised by purely sequential but rather iterative sequences of activities. The procedure model can be adjusted to respective conditions and constraints, by combining those steps in the needed order. Within the described design steps methods are provided to perform required tasks (Lindemann 2007).

“VDI 2206” is a more comprehensive design methodology, as it is suited for the development of mechatronic systems. It augments different domain-specific guidelines by methods for system design, system integration and property validation. The illustration as well as the approach originate from the V-model, which is used mainly in software engineering. This model describes the macro-cycle and thus the general process of mechatronic product development, which is traversed during the concretisation until series production (VDI-Gesellschaft Entwicklung Konstruktion Vertrieb 2004).

The systems engineering process is a comprehensive, iterative and recursive problem solving process which is suitable for the development of mechatronic products. In contrast to the procedure models discussed before it focuses on systems that do not only consist of products, but also people and processes. First step is the analysis of customer requirements in order to derive functional and performance requirements. Subsequently, functions are identified, decomposed and allocated to the requirements. During synthesis the product is defined in terms of physical and software elements and afterwards verified against the requirements. Accompanying activities and methods make sure that progress is measurable, alternatives are evaluable and decisions are documentable (Department of Defense 2001).

All presented procedure models have in common, that they help coping with complexity in product development. Furthermore cooperation between different domains or departments is promoted through their application (Lindemann 2007). In most of the described design steps models are created (e.g. requirement model, functional model, structural model), which represent different development perspectives on the product or system. Frequently the creation of one partial model has an impact on several other models. In this way, e.g. the functions of the product or system are derived from the requirements, which have been defined before. Since this usually happens implicitly, the dependencies between the partial models are not documented. This results in great efforts to restore consistency between the models if changes occur.

Maurer (2007) addresses this problem by proposing the use of matrices for the connection of the partial models. He differentiates between Design Structure Matrices (DSM), Domain Mapping Matrices (DMM) and Multi Domain Matrices (MDM), which are a combinatorial advancement of the first mentioned. Goal of his studies is to analyse, control and improve the dependencies of complex systems. For this purpose he suggests a number of analysing techniques that allow for the identification of connected structures by rearranging matrices or the tracing of impact chains to become aware of the implications changes might have. For visualisation purposes he uses matrices or strength based graphs, if the number of connected elements becomes too large. Both visualisations can be derived from each other and are included in the commercial software Loomeo (Maurer 2007), (Teseon 2009). Based on this preliminary work, Diehl, Hellenbrand and Lindemann developed the tool 3D-MDM, in order to visualise Multi Domain Matrices (MDM) three-dimensionally. By connecting it to Loomeo, two dimensional positioning and connection information can be reused to graph the different models on semi transparent layers (Diehl, Hellenbrand, and Lindemann 2008).

A very similar approach to connect product development information is followed by Sitte and Winzer in their method to develop mechatronic products, called Demand Compliant Design (DeCoDe). Likewise, they use matrices to connect requirements to the three complementary views functional view, structural view and process view (Sitte and Winzer 2007).

All described approaches to connect partial models of a product are very valuable. But as

Maurer observes, the collection of the dependencies as well as the achievement of high connection quality (avoidance of wrong and missing dependencies) still poses high challenges for the actual use of these methods (Maurer 2007). Present answers to this problem, like the use of interdisciplinary workshops or the collection of existing data (e.g. QFD, TRIZ, PDM), still imply high efforts and are automatable only in exceptional cases. For this reason it is necessary to further integrate the creation of connections in the development process limiting the additional work effort required to a minimum and not implement it as a subsequent procedure.

Important preliminary work, which is the methodological groundwork for the approaches described later in this paper, has been accomplished in the BMBF project “integrated virtual product creation” (iViP). The result was a software prototype “Function Oriented Design” (FOD) for the integrated support of requirements and functional modelling as well as the definition of product structures. The dependencies between these partial models can be expressed as qualitative or as quantitative parametric connections (Baumann, Kaufmann, Leemhuis, Kühn, Ragan, and Swoboda 2002), (Leemhuis 2002).

During the BMBF project MIKADO the FOD prototype, which is marketed by the CADsys GmbH, has been further enhanced by aspects of mechatronic product development and by methods of continuous verification of product maturity. To do so, functional mock ups have been developed to carry out complex verification tasks of mechatronic assemblies (Figge, Hajtmar, Hegewald, Rothenburg, and Swoboda 2008, 726-731).

Field of Application and Challenges

Based on the methods developed in the projects iViP and MIKADO the prototypical software Connection-Modeller is implemented in ISYPROM to support the early phases of Systems Engineering.

Field of application of the Connection-Modeller. Based on VDI 2206, the field of application of the Connection-Modeller is in the phase of system design (Figure 1).

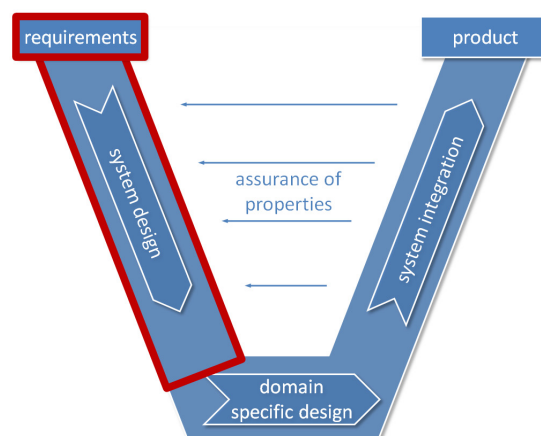


Figure 1: Field of application of the Connection-Modeller (highlighted) in the V-Model based on VDI 2206 (VDI-Gesellschaft Entwicklung Konstruktion Vertrieb 2004)

Requirements that have been specified before are the input for this phase. Based on them, the overall function of the mechatronic product is derived and further detailed into subfunctions. In a subsequent step the subfunctions are connected to a function structure in order to describe the behaviour of the mechatronic product and detect inconsistencies in an early stage. For each subfunction operating principles or solution elements are identified and further concretised to solution variants before the domain specific design is started (VDI-Gesellschaft Entwicklung Konstruktion Vertrieb 2004).

System Design today. In today's system design, the before mentioned process steps that are needed to define the overall system architecture, are handled isolatedly (Figure 2).

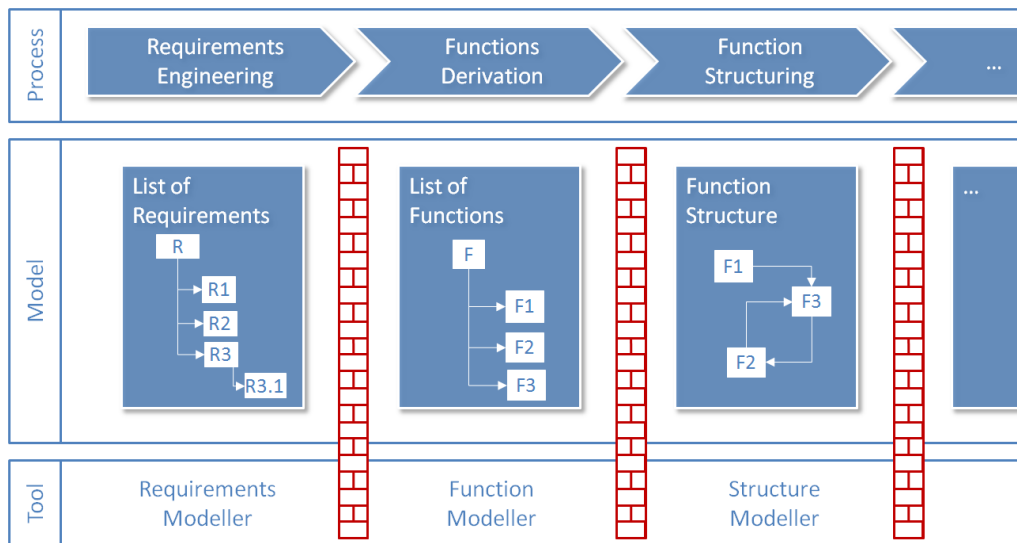


Figure 2: Process steps according to VDI 2206 with regards to the tools used and the models created.

The specified list of requirements is developed and managed mainly in a proprietary requirements modeller. Even though it is the input for the list of functions, the dependencies between the two lists usually get identified by the developer only implicitly and not documented at all. This mainly results from the fact that different tools are used for their development and management.

In the next step, the structuring and linkage of single functions with material, energy and information flows to generate a function structure follows a similar approach. Since the function structure is developed and managed in the structure modeller the functions need to be copied from the function modeller. In doing so, the dependencies between function structure and list of functions are not established and all information linked to elements of the list get lost when building up the structure (Figure 2).

In execution, this procedure results in two problems. When sequentially creating the described models, elements and structures that have been developed before cannot be re-used but need to be modelled again (see example: development of the function structure from the list of functions), because of the lack of continuity in software communication. This results in considerable additional expenses. Furthermore due to the missing connections between the models, changes made to one model create inconsistencies between the different models that require great effort when being removed.

Connection-Modeller for cross-domain dependency modelling

General Approach to connect cross-domain partial Models. In order to cope with the problems, that occur when partial models of a system are treated in an isolated way, the general approach of the Connection-Modeller provides means to define cross-domain connections between different partial models (Figure 3). These qualitative connections represent the general dependency between elements and can be further detailed and explained with attributes and comments. By demonstrating interdependencies between models of different domains the development process consistency of a system will be improved. Domain-specific connections,

like dependencies in-between requirements in the list of requirements or functions in the function structure, are still modelled in proprietary software tools, which are well-known to the respective developers, but can be used for further analysis in the Connection-Modeller later on.

Given a situation when a developer has to modify the geometry of a housing he will be provided with the information about requirements, such as mating dimensions, existing for this housing in order to check their compliance regarding the changes made.

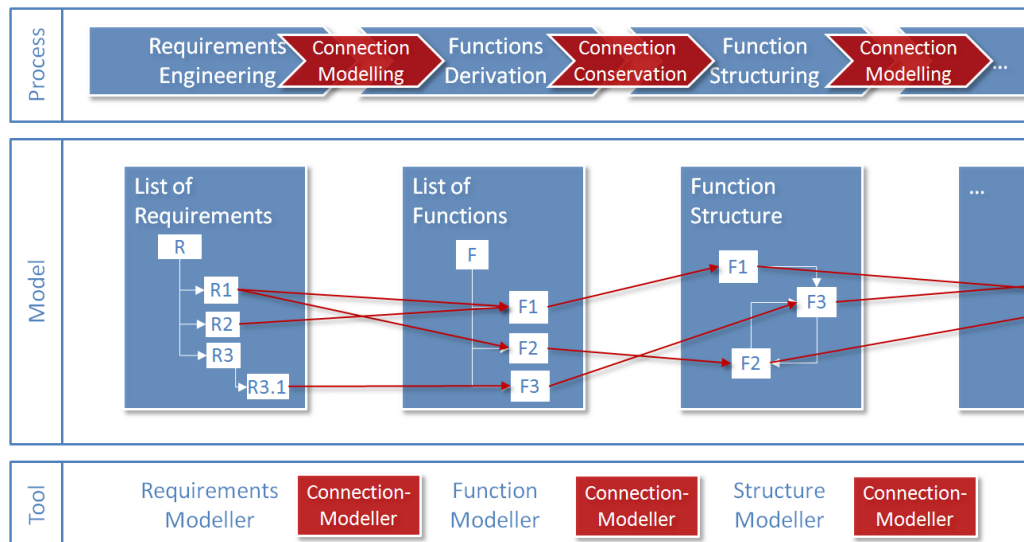


Figure 3: Process steps according to VDI 2206 enhanced by the use of the Connection-Modeller.

The Connection-Modeller acquires the models that are supposed to be connected to each other from the specific authoring tools and provides them to the user. For demonstrational purposes during the ISYPROM project requirements and UML models are supported and an interface to a PLM-System is implemented. Basically any models can be imported and used in the Connection-Modeller if an interface is implemented (Figure 4).

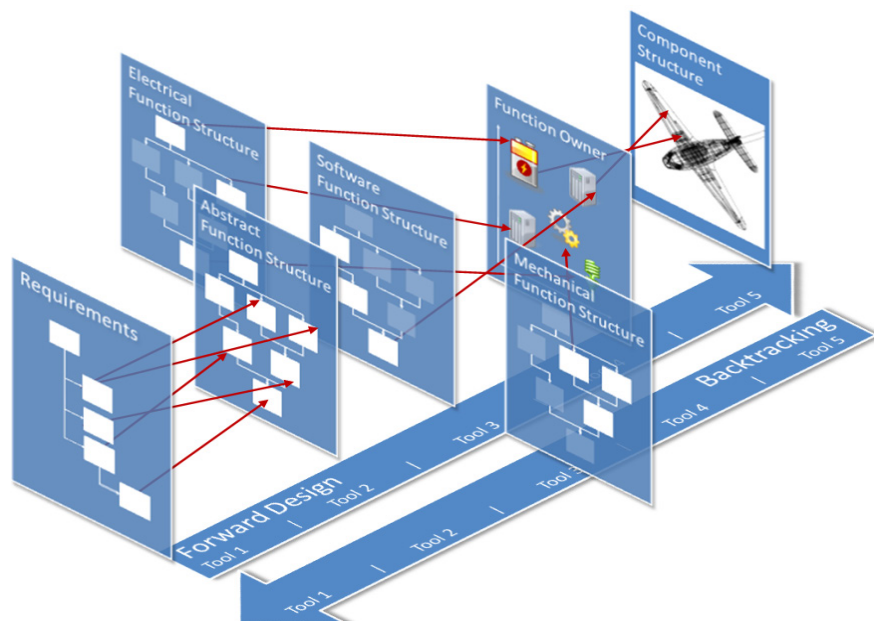


Figure 4: General approach to connect cross-domain partial models

This general approach allows for an easy adaptation and thus the support of different industrial use cases instead of only very specific ones. The Connection-Modeller saves only information regarding connections and references to the connected models. With every start of the Modeller, the partial models are loaded from proprietary databases, guaranteeing the actuality of the data.

As mentioned before, especially the creation of dependencies is one of the biggest challenges and the main hindrance for an industrial application of methods to connect information. Maurer (2007) proposes the use of interdisciplinary workshops with several developers for the collection of the dependencies, but observes at the same time, that it is a very time-consuming process. Following this approach, in ISyProM, two ways to create connection information become implemented: “Conservation of Connection Information” and “Intuitive Evaluation of Connection Information”.

Even though these functionalities are a first step towards a better integration in the development process, the objective for further research has to be the automation of dependency collection, since manual collection is too costly.

Conservation of Connection Information. In case of the conservation of connection information, a model has to be available in the Connection-Modeller. Elements of this model are provided to a different authoring tool by the Connection-Modeller where they are refined and recomposed to a new model. By providing the elements it is ensured that their unique IDs are retained. If this new model is loaded in the Connection-Modeller, connections between the original and the new model will be set automatically by matching the unique IDs of the elements.

For example, this could be functions in a list of functions (Figure 5 - step 1) which are provided to a authoring tool (Figure 5 - step 2) to further connect them to a function structure (Figure 5 - step 3). In doing so, the connections between list of functions and function structure are defined automatically without any adjustments (Figure 5 - step 5) in the Connection-Modeller. Furthermore there is no more need of creating the elements of the function structure manually in the respective authoring tool, helping to save time and avoid errors as the main benefits.

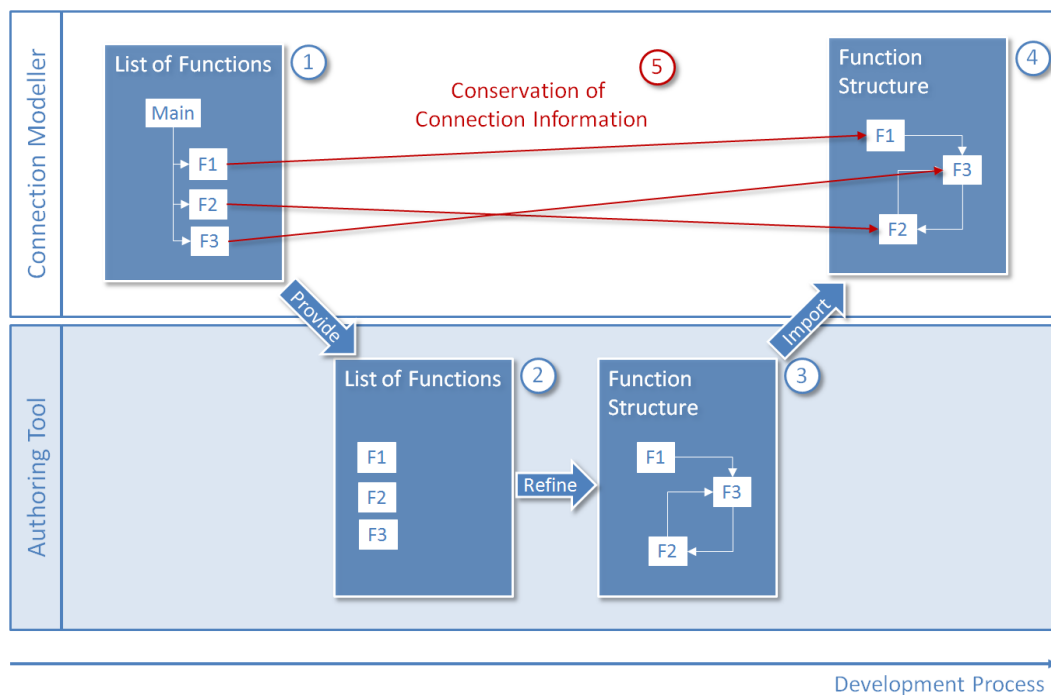


Figure 5: Automatic creation of connection information

Intuitive Evaluation of Connection Information. The methods for manual creation of connection information are similar to the Domain Mapping Matrices (DMM) suggested by Maurer (2007). There are always two partial models that are examined and analysed for dependencies on element level. Instead of matrices, like in FOD, adjacent tree structures are used for visualisation purposes. It is assumed that several developers are involved in the definition of connections, all with their specific technical discipline in mind. Developers with the according authorisations will be given the opportunity to set new connections, whereas all developers can flag any connection to immediately express their doubts regarding the correctness of a specific connection. A rating system is being implemented counting the expressed doubts. If several developers mistrust a certain connection the evaluation status of this connection will be set to yellow or red (Figure 6), which makes a review necessary. For example, in Figure 6 the connection between Requirement R3 and Function F1 is likely to be incorrect. This approach allows for identifying erroneous connections with little effort, generating time-savings as well as quality and actuality improvements through user integration.

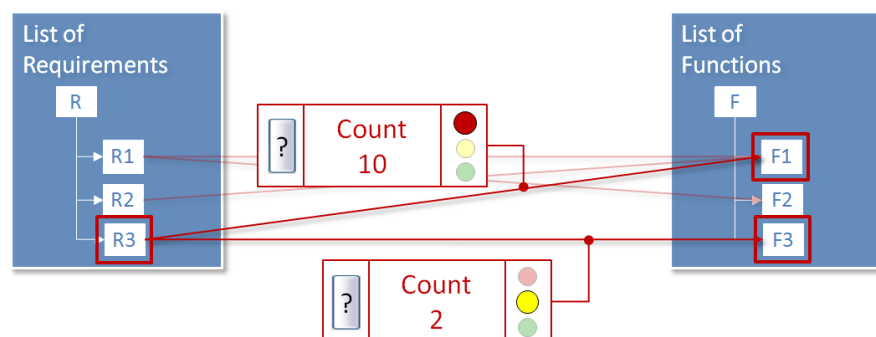


Figure 6: Evaluation mechanism for identification of erroneous information through user integration

Identification and Visualisation of Changes. The successive connection of models in the development phase establishes a continuous relation between e.g. requirements and component structures, which can be a major advantage when changes occur. The identification of model elements that have been changed or added since the last process step is automatically supported by the Connection-Modeller. These identified elements are highlighted to the user (Figure 7).

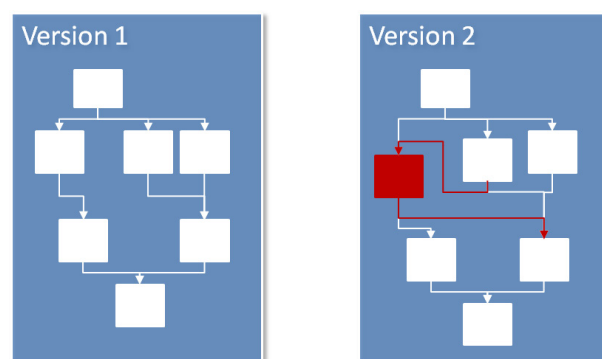


Figure 7: Identification and visualisation of changes

With the means of information reduction (see subsection “Demand-based Information Reduction”) as well as on the basis of the feed-forward-analysis (Maurer 2007), functionalities

become implemented to analyse cross-domain implications of detected changes on directly and indirectly connected elements. Besides the connection information the role of the user is taken into consideration and thus only changes that are relevant for the user become visualised.

As an example one can imagine that a customer modifies a requirement in order to receive a water-proof housing as the system will also be used for external applications in the future. When using the Connection-Modeller all developers working on the housing will get notified about this modification in requirements.

Proactive Quality Management. Part of the development of the Connection-Modeller is to define and evaluate simple constraints and rules, to verify models for completeness and other rules on the level of connections, objects, and metadata. The aim of this concept is to lay the foundation for an integrated proactive quality management where development rules can be deposited and checked for compliance. This allows for the detection of erroneous trends already in the early stages of product development where countermeasures can be initiated with significantly less effort.

As an example, the number of functions per control module can be monitored and a warning displayed if a certain number is exceeded. In case of safety-critical functions attention can be paid to the fact that these systems, which need to be realised redundantly, contain only few non-critical functions to reduce avoidable costs. This approach is meant to support error avoidance and thereby improve the quality of a system in a conceptual stage of the process.

Demand-based Information Reduction. Not only the quality of data influences the efficiency in the development of systems but also the amount of information provided. An oversupply hinders the selection of relevant information and thus delays the development process. To improve efficiency in system development, methods are developed to implement a demand-based information reduction, in order to provide developers only with information, which are relevant in the context of their work.

The provision of information on the basis of a demand-based approach ensures a higher relevance of the information provided, and in that manner, enhances the efficiency in the further processing of information. Demand-focused models help to automatically provide system developers with relevant model elements. For example, someone who is simulating safety issues of a chassis in a crash situation does not necessarily need to see requirements that call for a certain number of cup holders in the cockpit.

All direct dependencies between partial models of different domains are managed in the Connection-Modeller helping developers to recognise which other elements will be influenced by their work. Only these models elements need to be visible and accessible for the developer (Figure 8).

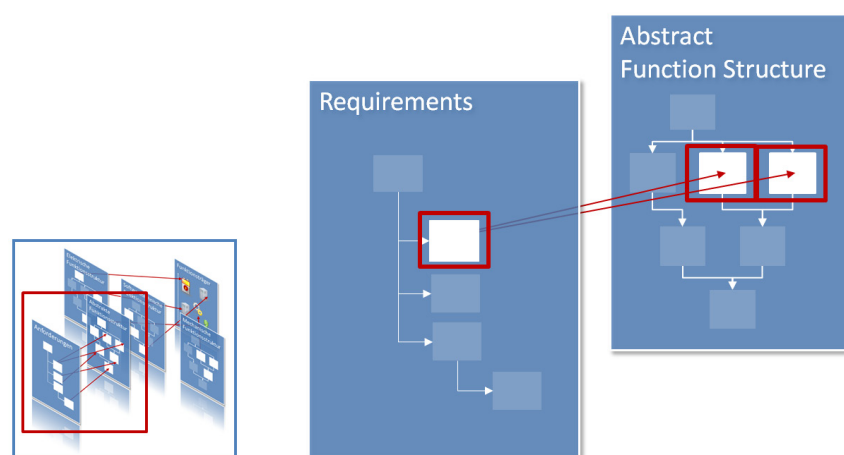


Figure 8: Provision of reduced information associated with a certain requirement

The reduction of information goes along with a concentration on the core activities of the developer, as his role is also a demand defining parameter. Additional information that can be read with the Connection-Modeller such as responsibility or maturity level of components will help developers to contact the right person in charge of any influenced element – easing coordination within companies and accelerating development processes.

Based on the demand, views can be created highlighting every element that has a direct or indirect connection to a specific selected model element. Elements that have no connection to the element of interest can be set invisible reducing the total amount of information significantly.

Utilisation of the Connection-Modeller

In the following, the utilisation of the Connection-Modeller shall be demonstrated in accordance to the procedure model VDI 2206 (see subsection “Field of application of the Connection-Modeller”). As a prerequisite at least two partial models need to be available whose elements can be connected. Starting from the requirements the overall function of a system is being derived and further detailed to a number of subfunctions forming a list of functions. All interdependencies between requirements and functions that have existed implicitly need to be explicitly modelled with the use of the Connection-Modeller. For this purpose both lists are opened in the Connection-Modeller and the connections between the according model elements are set. This activity is executed by experts for the according subsystem. When developers identify a connection in their subsystem that they consider doubtful they are given the opportunity to flag this connection with an intuitive evaluation mechanism. All connections that have been doubted will be rechecked in decreasing order of their counter (see subsection “Intuitive Evaluation of Connection Information”) to steadily improve the quality of the managed connections.

Once the definition of the connections has been completed a function structure needs to be developed. For that purpose all elements forming the list of functions get provided by the Connection-Modeller to a structure modelling tool, where the according flow of information, energy and material get modelled depending on the processing sequence. There is no more need to model the connections between list of functions and function structure afterwards as all connections are conserved and established automatically when opening the structure with the Connection-Modeller. This procedure continues until the architecture of a system is fully defined and all partial models are connected according to the sequence of process steps described in VDI 2206.

During the definition of connections a proactive quality management supports the developers by providing context-based suggestions for connections and evaluations regarding beforehand defined design rules.

Additionally, it is feasible to visualise model changes in models opened in the Connection-Modeller. Prerequisite is that the models which have been modified in comparison to the last status of work, were connected to another model with the Connection-Modeller before. These elements will be highlighted, which together with the demand-based information reduction allows for an easier impact analysis in the change management process.

Summary and Outlook

The present paper shows concepts for a software prototype to solve problems and deficits in current product development processes and methodologies, especially for mechatronic product development as described in the state of the art. Advanced solutions to manage product model information like requirements, functions and product structures by cross-domain connection of these partial models are being developed. The key concept is the creation and management of these relations outside of proprietary software tools using neutral representations in standardised data models. The concept presented enables the connection of all kind of information and is not restricted to requirements, functions, etc. New approaches for the conservation and manual creation of connections between domain-specific models are being developed to support this process efficiently.

The continuous connection between all partial models in the product development process, starting in early phases, enables an efficient management of interconnected domain-specific development processes. Changes can be propagated efficiently and implications based on dependencies between partial models can be detected easily. The connections form the basis for context-based analysing and assessing of the product architecture in a very early development phase. The developed concepts provide functionality for pro-active quality checks like completeness and integrity of connections between models (Table 1).

Table 1 Functionalities and benefits of the Connection-Modeller

Functionality	Benefit
General Approach	
Connect cross-domain models	Improve consistency of a system
Forward Planning & Design	
Conservation of connection information	Time-savings & error avoidance
Intuitive evaluation of connection information	Time-savings & quality improvement by user integration
Pro-active quality management	Error avoidance & quality improvement
Demand-based Information Reduction	Improve efficiency by information reduction
Back Tracking	
Identification and Visualisation of Changes	Actuality of data & impact propagation

Intermediate project results have raised several topics for further research activities. A major problem in modelling and managing requirements, functions and other models of a real product is caused by the enormous amount of elements and their interdependencies. State of the art tools mainly use tables, lists and tree structures in a 2D environment. For a user this kind of interaction with a software tool is neither very appropriate nor intuitive and therefore rather fault-prone. For that reason big potential can be seen when transferring the technology into a 3D environment (e.g. virtual reality) or on multi-touch tables. The intuitive interaction mechanisms realised with these technologies and the improved visualisation of models and their connections provide a significant benefit. In the further course of the project it will be investigated how dependency modelling can be realised in these environments and according interfaces to the Connection-Modeller will be developed.

A Digital Functional Mock-Up, as a logical extension of the Geometric DMU, is targeted to represent and investigate a product's function, its behaviour and operating performance (Krause, Rothenburg, Wöhler, and Romahn 2005). The execution of FMU analyses requires

the existence of functional descriptions and simulation models. To perform functional simulations in early phases of product development, the concepts and methodologies of FMUs are to be extended to these phases. The advanced concept of a development accompanying FMU must consider the maturity of available models. It is subject to research to investigate which kind of analysis can be performed on the basis of immature information and which validity such results have. Maturity of results can be improved by using pre-defined modules and results from past projects. The Connection-Modeller holds the opportunity for steady enrichment of connection information along the development process supporting continuity. In doing so models with different levels of maturity can be included in the process of dependency modelling allowing developers to run initial analyses in the early phases of system development.

BIOGRAPHY

Prof. Dr.-Ing. Rainer Stark is head of the Division Virtual Product Creation and director of the Chair for Industrial Information Technology at the Technische Universität Berlin since 02/2008. After his studies in mechanical engineering (Ruhr University Bochum and Texas A&M University) he received the Dr.-Ing. degree from the Universität des Saarlandes Saarbrücken. During his industrial activities of many years he worked in different leading positions in the automotive industry. His research topics are the intuitive and context-related information modelling, intuitively usable and functionally experienceable virtual prototypes, the function-oriented Virtual Product Creation as well as development processes and methodologies for the product modelling.

Dipl.-Ing. Grischa Beier studied mechanical engineering with a specialisation in engineering design (Technical University Ilmenau - Germany, Universidade Federal de Santa Catarina – Brasil, St. Petersburg State University of Information Technologies, Mechanics and Optics – Russia). Since 2006 he works as a research fellow at the Fraunhofer Institute for Production Systems and Design Technology, Division Virtual Product Creation. His research interests include modelling procedures in product creation and information provision in cross-domain development processes.

Dipl.-Ing. Asmus Figge, studied mechanical engineering with a specialisation in engineering design at the Technische Universität Dresden. Since 2008 he works as a research fellow at the Chair for Industrial Information Technology at the Technische Universität Berlin. His research interests include function oriented design in mechatronic development processes.

Dipl.-Ing. Thomas Wöhler studied information technology in mechanical engineering (computational engineering sciences) with a specialisation in engineering design and manufacturing at the Technische Universität Berlin. After the completion of his Dipl.-Ing. degree in 1997 he worked at the Technische Universität Berlin at the Chair for Industrial Information Technology. Since 2004 he works as a research fellow at the Fraunhofer Institute for Production Systems and Design Technology, Division Virtual Product Creation. His research interests include geometric modelling, especially modelling of freeform surfaces, and the adaptation of geometric models for functional validation of virtual prototypes.

Acknowledgements

This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Framework Concept "Research for Tomorrow's Production" (funding number 02PC105x) and managed by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the contents of this publication.

References

- Baumann, R., Kaufmann, U., Leemhuis, H., Kühn, T., Ragan, Z., and Swoboda, F. 2002. Funktionsorientiertes Entwerfen. In Krause, F.-L., Tang, T., Ahle, U. (Hrsg.). *Integrierte Virtuelle Produktentstehung – Abschlussbericht*. Stuttgart: Fraunhofer IRB-Verlag.
- Department of Defense 2001. *Systems Engineering Fundamentals*. Virginia: Defense Acquisition University Press.
- Diehl, H., Hellenbrand, D., and Lindemann, U. 2008. Transparent 3D Visualization of mechatronic System Structures. In International Design Conference – Design 2008 (Dubrovnik, Croatia).
- Figge, A., Hajtmar, M., Hegewald, T., Rothenburg, U., Swoboda, F. 2008. Integrierte funktionale Validierung mechatronischer Produkte mit FMU. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Jg. 103, H. 10, 726–731.
- InMediasP 2009. <http://www.inmediasp.de/>.
- ISYPROM 2009. <http://www.isyprom.de/>.
- Krause, F.-L., Rothenburg, U., Wöhler, Th., Romahn, A. 2005. Functional Mock-Up. In: ProStep-iViP-Verein (Eds.), Krause, F.-L., Rothenburg, U. (Eds.). *Engineering 2010: Setting the Course for Innovative Engineering*. Augsburg: vmm wirtschaftsverlag.
- Leemhuis, H. et al. 2002. Function Oriented Product Modelling Based on Feature Technology and Integrated Constraint Management. In *Proceedings PDT Europe 2002* (Turin, Italy).
- Lindemann, U. 2007. Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden. Berlin Heidelberg: Springer-Verlag.
- Maurer, M. S. 2007. Structural Awareness in Complex Product Design. In *Produktentwicklung München Band 56*, München: Dr. Hut Verlag.
- Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H. 2007. *Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung*. Berlin Heidelberg: Springer-Verlag.
- Karlsruher Institut für Technologie (KIT) 2009. <http://www.produktionsforschung.de/verbundprojekte/index.htm>.
- Sitte, J., Winzer, P. 2007. Methodic Design of Robot Vision Systems. In Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation (Harbin, China).
- Teseon 2009. Loomio. <http://www.teseon.com/>.
- Ulrich, H., Probst, G. 1995. Anleitung zum ganzheitlichen Denken und Handeln. Ein Brevier für Führungskräfte. 4. Aufl., Bern Stuttgart: Haupt Verlag.
- VDI-Gesellschaft Entwicklung Konstruktion Vertrieb. 1993. VDI 2221 – Systematic approach to the development and design of technical systems and products. Düsseldorf: VDI-Verlag.
- VDI-Gesellschaft Entwicklung Konstruktion Vertrieb. 2004. *VDI 2206 – Design methodology for mechatronic systems*. Berlin: Beuth Verlag.