INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

VISUALIZATION OF INTERDISCIPLINARY FUNCTIONAL RELATIONS IN COMPLEX SYSTEMS

L. Becerril, D. Kasperek, M. Roth and U. Lindemann

Keywords: visualization, interdisciplinary functions, complex systems

1. Introduction

Mechatronic systems include an increasing amount of electronics and software, which leads to increasing complexity. The design of these systems requires the collaboration of experts from several disciplines [Alvarez Cabrera et al. 2009]. However, especially in the conceptual design phase, there is a critical deficit of methods and tools to support this interdisciplinary development processes [Follmer et al. 2010, 2011]. This often causes inefficient development and change-management processes.
According to Rui-qin and Hui-Jun [2004], multidisciplinary systems often lack standardized and commonly used representing methods. Instead systems are often illustrated by a group of single-discipline diagrams, which in many cases do not have the same level of abstraction. Stark et al. [2010] argue that an approach that focuses on the modelling of cross-domain dependencies is required in order to support product developers in analysing and controlling complexity and the interdependencies between their existing models [Stark et al. 2010].
In order to manage complexity and to have an efficient development and configuration process, mechatronic systems need to be seen as integrated systems instead of a group of single-discipline subsystems. System designers need to visualize the interdisciplinary relationships among system's elements, while maintaining a bird's-eye view of the main system's functions [van Beek and Tomiyama 2008a]. The current shortage of tools supporting this activity is the motivation for this work.
Functions represent a system at several levels of detail, which allows changing the level of abstraction while preserving the model's consistency [Alvarez Cabrera et al. 2009]. Depending on the level of abstraction, functions can be expressed solution neutral, and hence discipline neutral. Therefore this paper focuses on a function-oriented representation of systems. Through the visualization of functional relations, complex systems can be represented in a common model for all involved disciplines. This contributes to the system understanding and traceability, and supports a better collaboration among engineering disciplines during the development process. As a result, companies can develop multi-disciplinary integrated systems instead of assembling single-discipline subsystems. Furthermore, modularizing and standardizing these integrated systems at a functional level improve the incorporation of new technologies, upgradability, and forward-compatibility.
The multi-perspective functional model, presented in this paper, is based on the Function-Behaviour-State approach (FBS) and on the Systems Modelling Language (OMG SysML), and visualizes and stores knowledge about the system and its architecture. This visualization concept can provide an overview of the system as well as, for example, represent the information flow between two specific functions, while maintaining the models consistency by providing tailored diagrams based on information from a digital database. These diagrams have different perspectives, levels of abstraction,

and number of details of the same system, and therefore, can support interdisciplinary design activities accurately at every development phase.
Moreover, this paper focuses on customizable mechatronic systems from the perspective of product development. The first part presents an overview of the existing visualization concepts. Then the methodology of this work is introduced. Afterwards, a new model for visualization of interdisciplinary functional relations is suggested and discussed. As the results of this work were developed in cooperation with an industrial partner out of the mobility sector, they are consequently evaluated within a case study on a complex subsystem of the company's product family. Finally, the concept and its application are discussed and an outlook for future works is provided.

2. State of the art

The system's architecture describes the system on many levels of abstraction, and specifies its elements and their functions [Rapp 2010], [Ponn and Lindemann 2011]. Van Beek and Tomiyama [2008a,b] describe the architecture of complex systems as a pyramid, with few abstract functional descriptions at the top and many component details at the bottom.

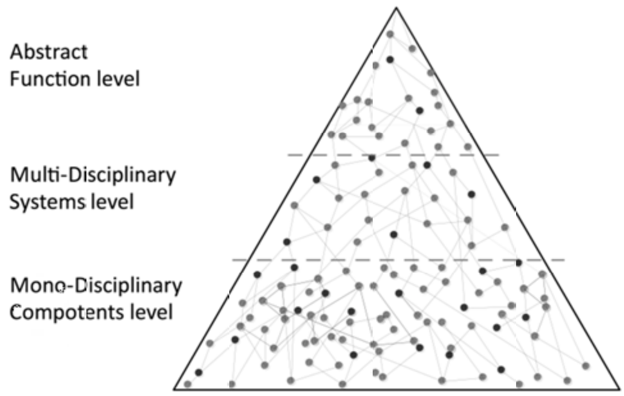

Figure 1. Pyramid representing the system's levels of abstraction [van Beek and Tomiyama 2008b]

Mechatronic systems integrate mechanical systems, electronic systems, and information technologies [VDI 2004]. They can be decomposed into functional subsystems across different levels of abstraction [van Beek and Tomiyama 2008a]. These subsystems generally are the executive mechanism, the sensing and testing subsystem as well as the information processing and control subsystem [Rui-qin and Hui-Jun 2004].
Rodenacker [1971] defines function as a relationship between input and output of information, energy, and material. In contrast, Sørensen [1999] defines function as an action desired to fulfil the systems purpose. In the context of this work, the authors define the term functions in the following way: Functions transform inputs to outputs in order to contribute to the system's purpose. Thus, functions can be connected to each other through inputs and outputs or through their influence on states of the system [Roth et al. 2013].
The different characterizations of functions have led to a variety of function-oriented representations of systems. Depending on the problem to address, different focuses and perspectives might be useful [Ponn and Lindemann 2011], [Herberg and Lindemann 2012]. In the context of this work, existing visualization concepts that focus on functional relations are analysed. Table 1 presents an overview of these concepts and their main properties.

Table 1. Existing visualization concepts for functional relations

Visualization concept	Description
Relation-oriented model	• represents interaction between functions: how they contribute to each other and to the system's purpose [Ponn and Lindemann 2011], [Herberg

	and Lindemann 2012] • classifies functions in useful and harmful functions [Ponn and Lindemann 2011]
Flow-oriented model	• represents flows (i.e. matter, energy, and signal flows) and their transformations caused by functions [Ponn and Lindemann 2011], [Herberg and Lindemann 2012] • elements are defined through in- and outputs [Kernschmidt et al. 2012]
Semi-formal specification of product functions	• represents functions and malfunctions, states and misstates, environmental influences, and functional objectives [Gausemeier et al. 2001] • functions are interrelated via states and misstates [Gausemeier et al. 2001] • transitions and functional requests relate functions with states [Gausemeier et al. 2001]
Symbolic representation	• represent function-families through solution-neutral graphical icons (e.g. checking, cutting) [Weyrich et al. 2011] • illustrates each subsystem (executive mechanism, sensing and testing, control) with a different graphical symbol [Rui-qin and Hui-Jun 2004].
Goal tree-success tree (GTST)	• represents the decomposition of functions [Modarres 1999] • merges the structural and functional hierarchy [Modarres and Cheon 1999] • the top level "functional objective" describes the purpose of the system [Modarres 1999] • the bottom level allocates the structure elements to their functions [Modarres 1999]
Function-Behaviour-State (FBS)	• represents the hierarchical structure of a system [Umeda et al. 1990] • comprises three layers: the function layer, the behaviour layer, and the state layer.[Alvarez Cabrera et al. 2009] • Functions are decomposed into sub functions until they can be associated with physical effects (behaviour layer), and are then associated with the physical components [Erden et al. 2008], [van Beek and Tomiyama 2008a].
Systems Modelling Language (OMG SysML)	• represents the structural composition of systems, their function-based behaviour and requirements [Friedenthal et al. 2011] • A model represents the whole system, while a diagram illustrates a point of view [Weilkiens 2011]. • The Block Definition Diagram defines the structural relationships of blocks (system elements) [Friedenthal et al. 2011]. • The Activity Diagram represents how actions execute based on their inputs, outputs (object flow) and logical relationships (control flow) [Friedenthal et al. 2011]. • Allocations describe the relations between behaviour and structure. However, they cannot be displayed graphically [Follmer et al. 2010].

3. Visualization concept

3.1 Requirements on the visualization concept

For the development of the multi-perspective functional model, research on requirements for visualization of complex systems was done. A summary of the findings is presented below.

The visualization concept has to provide a system overview and support representing different levels of abstraction together with the decomposition of the system into subsystems [van Beek et al. 2008a], [Wölkl and Shea 2009]. Hence, it needs to represent the system's elements at different levels of abstraction, including functions and entities [van Beek and Tomiyama 2008a], [Follmer et al. 2010]. Entities are all physical objects in the system [van Beek et al. 2010]. Additionally, the representation of the relationships between these elements is essential [van Beek and Tomiyama 2008a], [Follmer et al. 2010] – especially between disciplines. Thus, Intra- and interdisciplinary relationships – such as

flows, logical and hierarchical relations – are required in the model as well as relationships with the environment (interfaces). Furthermore, the visualization needs to be understandable, also for a person who is not familiar with the system. This comprises discipline neutral and graphical representation and intuitive navigation through the system [Alvarez Cabrera et al. 2009], [Gausemeier et al. 2009]. In order to manage the required amount of information, the concept must be computer interpretable [Wölkl and Shea 2009]. Finally the industrial partner requires expandability and the possibility of re-use of partial models as well as a modular configuration and standardized interfaces.
The requirements from the literature were complemented with requirements of the industrial partner (marked with an asterisk in Figure 2), which were specified in workshops, in order to increase the model's acceptance and practical applicability. Furthermore, the requirements can be classified into three groups: structure and views (regarding the configuration of the model itself), content, and usability. The overview of the main requirements on the visualization concept and their classification are documented in Figure 2.

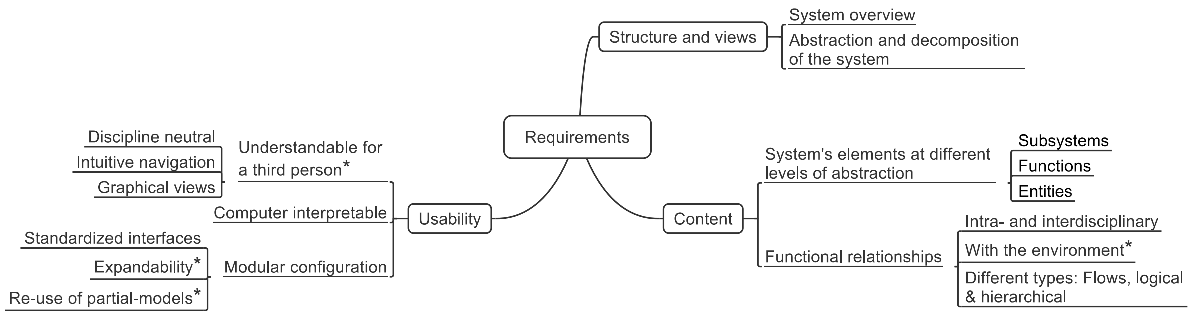

Figure 2. Classified overview of the main requirements for the visualization concept

3.2 Assessment of existing concepts

The assessment presented below is based on weaknesses and advantages stated in the literature. Additionally, an exemplary subsystem of the industry partner was modelled in each of the visualization concepts from Table 1, and issues during the application were identified regarding its practical applicability.
Table 2 presents a summary of the assessment results. The symbols "-", "o", and "+" graphically represent the fulfilment of the requirements by each concept. The symbol "-" represents the case when the visualization concept is not able to model a certain component (e.g. flows) or does not fulfil a certain requirement at all. The symbol "o" represents a moderate fulfilment and/or restricted applicability. Furthermore, "+" corresponds to a good fulfilment of the requirement and applicability.

Table 2. Assessed visualization concepts

	Relation oriented	Flow oriented	Semi-formal sp.	Symbolic rep.	GTST	FBS	SysML
Structure and views							
System overview	-	-	-	-	o	+	o
Flexible level of abstraction	o	o	o	-	+	+	+
System decomposition	-	-	-	-	o	+	+
Content							
Functions	+	+	+	+	+	+	+
Entities	-	-	-	-	o	o	+
Intra-disciplinary	+	+	+	o	+	+	+
Interdisciplinary relationships	o	o	o	+	o	o	o
Relationships with the	-	+	+	-	-	-	+
Flows	-	+	+	+	-	-	+
Logical relationships	+	-	+	o	+	-	+
Hierarchical relationships	-	-	-	-	+	+	o
Usability							

Discipline-neutrality	+	+	+	**o**	+	+	+
Easy use and navigation	+	+	+	**o**	**o**	+	**o**
Graphical representations	**o**	**o**	**o**	+	**o**	+	**o**
Computer interpretability	**o**	**o**	**o**	**o**	**o**	**o**	+
Modular configuration	**o**	**o**	**o**	**o**	**o**	**o**	+

Even though none of the concepts fulfils all requirements, the assessment results show that the most adequate concepts are FBS and SysML. FBS illustrates the hierarchy within functions and relates the lowest level ones with physical components by means of behaviours. SysML represents a number of views of the same system in different diagrams. The assessment scores of FBS and SysML are not decisive enough for a final choice, especially since the assessment is merely qualitative. On one hand, FBS models connect different levels of abstractions and therefore support system understanding, but do not represent relationships within the same level of abstraction. On the other hand, SysML visualizes these relations among functions and among system components on the same level of abstraction through flows, but cannot represent connections between elements of different types graphically. As both features are essential an integrated solution for these conflicting features is needed.

3.3 Visualization by multi-perspective functional model

The multi-perspective functional model is based on the FBS model and on SysML. It integrates the perspective of both models towards the system. It is a function-oriented model with focus on interdisciplinary relations within complex mechatronic systems and proposes a representation of the system based on the pyramid by van Beek and Tomiyama [2008a]. The model consists of a three-dimensional pyramid that illustrates, the hierarchical relationships among functions and entities (similarly to FBS) in the vertical direction and the flows among them (similarly to SysML) in the horizontal direction.

3.3.1 Structure of the model

The multi-perspective functional model structures mechatronic systems in three layers:

- Functions
- Elementary functions
- Structure

In place of the layer “behaviour” from the FBS model, the layer “elementary functions” links the functions with the structure. Elementary functions are functions that cannot be further decomposed; they often are not solution-neutral. The difference between elementary function and behaviour is that behaviour can only describe physical changes in the system, while elementary functions do not need to be associated with a physical entity. This increases the flexibility of the model – for example in the context of modelling software.

In contrast to the FBS model, the multi-perspective functional model supports two different types of relationships. It not only represents the hierarchical decomposition (e.g., functions and sub functions) but also the flows (control, energy, material, information), which are based on the SysML language. Therefore, each element of the system can be linked with elements from the same layer (trough flows) or from the adjacent ones (through hierarchies). Each modelled system can be combined and linked as a module to form larger systems.

Since the main focus of this work is customizable systems, the multi-perspective functional model incorporates the customer or system user view to the representation of the system architecture through features. A feature is defined as “a characteristic of a product with customer value” and can be linked to functions (i.e. functional feature) or to entities (i.e. non-functional feature). Features describe the added value of the elements in the system; hence they can improve design traceability with regard of allocating system functions and components to the customer’s requirements.

Figure 3 illustrates the pyramidal system model with features, functions and structure layers and the relationships among its elements. Relationships between elements of different layers can be described by the verb “realize”: entities realize elementary functions, which, at the same time, realize functions of higher levels of abstraction. Moreover, features are linked to the other elements by a "fulfil" relationship.

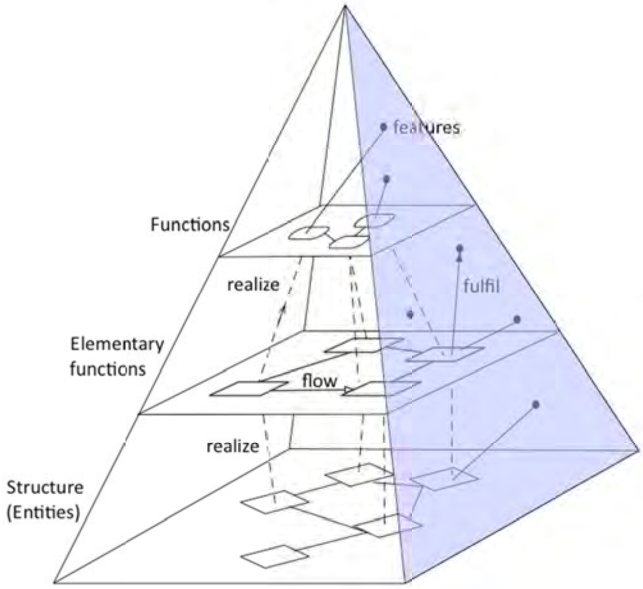

Figure 3. System model with features, functions and structure layers

3.3.2 Sectional views and use-cases

The three-dimensional representation of the system allows both, to examine the hierarchal relationships and the information flows using one and the same model and thus ensuring consistency. The "cross-section" represents the flows within the system (horizontal view) and the "transversal section" of the pyramid (vertical view) represents the hierarchical decomposition.

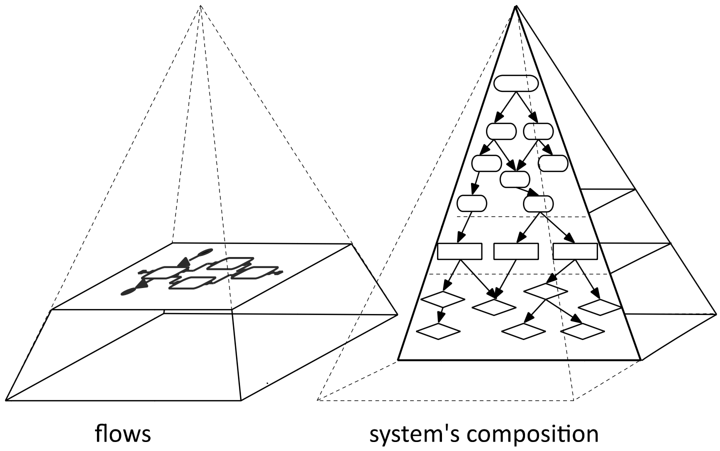

Figure 4. Transversal and cross-section of the system

The horizontal views can be represented in a SysML activity diagram, or in a SysML internal block diagram at the structural level. The vertical view illustrates the system's decomposition from the top-level functions to the physical components with a Feature-Function-Structure diagram – similarly to a FBS diagram. The different views provide tailored diagrams for specific design tasks. Use-cases are defined in order to identify the different design tasks that could be supported by the model. A total of 60 use-cases are identified and analysed in regard of what information is necessary to generate the corresponding diagrams based on the whole model. Furthermore, depending on the information to illustrate, the resulting diagrams are assigned – for example an FFS diagram for a vertical view. Three use-cases are exemplary described in Table 3.

Table 3. Use-cases of the model

Use-case	View	Layer(s)	Flow(s)	Constraint(s)	Result
How does function α exchange information with other functions	horizontal	functions	object flow (information)	function α and related functions,	SysML activity diagram
What sub functions does function α and function β have in common?	vertical	functions	-	functions related to function α and function β	FFS diagram
Which components fulfil which features?	vertical	features, functions, elementary functions and structure	-	-	FFS diagram

3.3.3 Implementation within the software tool

The Multi-perspective functional model is implemented using the software tool *Soley*. *Soley* is an engineering software for the formalization and computational application of knowledge [Soley-technology 2014]. It provides a schematic or network-like representation of structures based on graph-grammars [Soley-technology 2014] and allows users to create their own metamodel, which is the main reason for its adoption in this work.

The software distinguishes two types of elements: nodes and edges. Nodes are the system elements and edges represent the relationships among these elements. In this paper, the nodes classes are features, functions, elementary functions, entities, and objects and the edges classes are: fulfil, realize, flow, and part of. Furthermore, in *Soley* nodes and edges possess attributes that describe and characterize the system elements they mirror. For the scope of this paper, the attributes considered are the element's name and an unique identifier; objects also have the attribute "flow type" in order to be classified according to the flow they represent (i.e. information, energy, material).

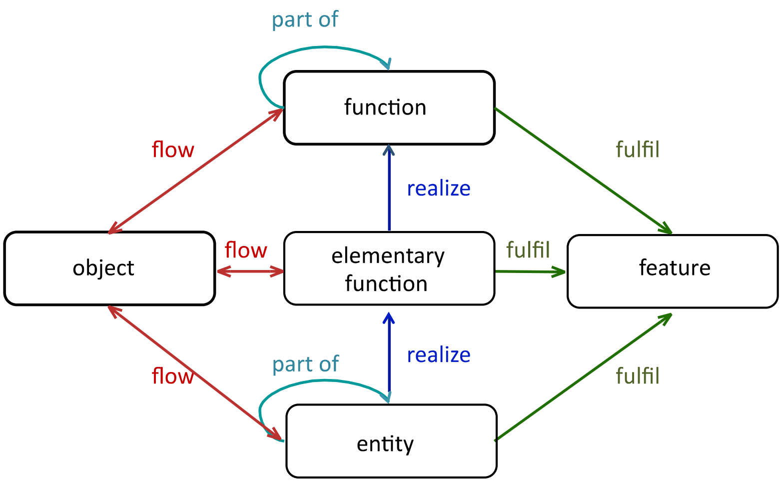

Figure 5. System elements (nodes) and their relations (edges) for implementation in *Soley*

An advantage of *Soley* is the possibility of adapting the model through sets of rules and sequences. Rules represent the activities the software needs to perform in order to create parts of the model, while sequences define which rules and in which order they should be executed. In other words, they describe which elements and relationships are displayed. By the means of sequences different parts of the system can be visualized in order to create the different views of the multi-perspective functional model.

For example, the sequence "create FFS-diagram" contains the rules for illustrating the vertical view of the system model – the FFS-diagram. These rules generate the necessary nodes (features, functions and entities) and link them through structural edges (i.e. "fulfil", "realize", and "part of"). For depicting the horizontal layers, the required rules generate the function nodes (for example in the function layer) and the object node, Then they connect these nodes by means of flow-edges. The result is similar to a SysML activity diagram. However, a benefit of the implementation in *Soley* is that elements can be "turned on and off" depending on their attributes and on their connections with help of constraints within the rules (see Figure 7). These constraints are presented in the use-cases, based on which additional diagrams that support specific design activities can be generated through further rules and sequences.

3.4 Application of the model

For its validation, the multi-perspective functional model is applied on a complex sub-system of the industrial partner's product family. First, the relevant types of elements and relationships are defined. Then, information about these elements is acquired based on available documents and drawings, as well as through expert workshops. The elements and their relationships are structured using DSMs and DMMs and the model is implemented in *Soley*.

Figure 6 illustrates the subsystem from the vertical point of view according to the use-case "Which physical components fulfil which features?" It depicts the entities, the elementary function, the functions, and the features. This view could support for example a product developer to transfer a feature from an existing product to the next generation by identifying the functions, which realize the electronic water level regulation and the related components. Those components could form a reusable module, or could be replaced if obsolete by reusing only the functional architecture.

As shown in Figure 6, the multi-perspective functional model – in combination with *Soley* – only depicts the relevant view, facilitating the development task by reducing the amount of information.

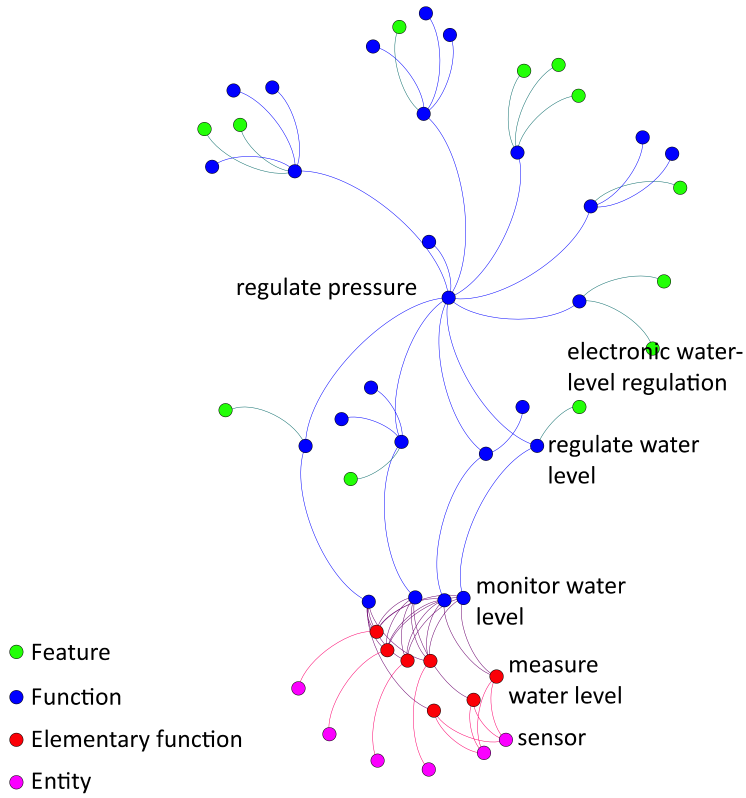

Figure 6. Exemplary view, generated out of the multi-perspective functional model for the use-case: "Which components fulfil which features?"

4. Benefits and drawbacks of the multi-perspective functional model

The assessment in Section 3.2 reveals that the FBS model can connect different levels of abstractions and therefore, support system understanding. Meanwhile, SysML visualizes the interdisciplinary relations among functions and among system-components through flows. The authors of this paper develop the multi-perspective functional model, which integrates both perspectives and hence, supports the visualization of interdisciplinary relations and system decomposition.

The layers in the multi-perspective functional model represent the levels of abstraction of the system. Through this visualization, the model connects features from the customer's perspective with functions from the systems designer's perspective, which are connected at the same time with functions from the components- or software-developer's perspective. Additionally, flows represent the relationships among functions and components within the same level of abstraction. This, along with discipline neutrality, supports the system understanding in all involved disciplines. Furthermore, the model supports design traceability as necessary for the management of the integration of new functions and features. With the generated model, functions can be clustered or grouped based on their attributes with, for example, matrix-based approaches in order to form modules for standardization and reuse.
Through the application of “rules” and “sequences” in the software tool, the model can generate tailored diagrams for each use-case and hence display only relevant information for its current user. Therefore, it is able to provide an overview of the system as well as, for example, to represent the signals between two specific functions. Nevertheless, the overview declines when a high numbers of elements and relations are represented in the same diagram. However, all known information about the system is stored in the digital model to ensure consistency.

5. Conclusion and outlook

This paper reviews and examines tools and methods that visualize interdisciplinary functional relations in complex mechatronic systems in order to support the development and configuration process. The literature review reveals a shortage of adequate visualization tools for interdisciplinary design tasks. Requirements for the desired visualization are defined and existing tools and methods are assed towards their degree of fulfilment of these requirements for visualization.
The developed multi-perspective functional model integrates FBS and SysML and depicts the system's composition at every level of abstraction and the interdependence among elements of the different disciplines as well as the realization of functions through physical components. It represents a common model that supports the illustration and understanding of interdisciplinary relations within complex mechatronic systems, and therefore, serves as a tool to support the collaboration of different engineering disciplines during the development of complex systems.
The multi-perspective functional model considers mechatronics as integrated systems. As a consequence, synergies that emerge during the development of complex systems are reinforced. Additionally, modularizing and standardizing the system as a whole supports the reuse of architectures at the functional level, which enhances the integration of new technologies, the upgradability, and the forward-compatibility of the systems.
Through the implementation in a software tool, knowledge about the system and its architecture is stored in one consistent model, while providing tailored diagrams for each design activity. In summary, the multi-perspective functional model, employed within the development and configuration processes, supports the collaboration among disciplines, increases system understanding and design traceability, and a more efficient knowledge reuse.
In the future, the model will be applied in a variety of systems. Special focus will be on systems with a large number of elements and high complexity. Furthermore, future studies can increase the detail of the represented elements, for example through more attributes in order to support the modularization task and reuse.

References

Alvarez Cabrera, A. A., Erden, M. S., Tomiyama, T., "On the potential of function-behavior-state (fbs) methodology for the integration of modeling tools", Proceedings of the 19th CIRP Design Conference, Cranfield, 2009, pp. 30-31.

Erden, M. S., Komoto, H., Van Beek, T. J., D'amelio, V., Echavarria, E., Tomiyama, T., "A review of function modeling: Approaches and applications", Ai Edam, 22, 2008.

Follmer, M., Hehenberger, P., Punz, S., Zeman, K., "Approach for the creation of mechatronic system models", International Conference on Engineering Design, Culley, S.J.H., B.J.; Mcaloone, T.C.; Howard, T.J.; Lindemann, U. eds. Lyngby/Copenhagen, 2011.

Follmer, M., Hehenberger, P., Punz, S., Zeman, K., "Using sysml in the product development process of mechatronic systems", International Design Conference, Marjanovic D., S.M., Pavkovic N., Bojcetic N. eds., Dubrovnik, 2010, pp. 1513-1522.
Friedenthal, S., Moore, A., Steiner, R., "A practical guide to sysml: The systems modeling language", pp. 2011.
Gausemeier, J., Flath, M., Mohringer, S., "Conceptual design of mechatronic systems supported by semi-formal specification", International Conference on Advanced Intelligent Mechatronics Proceedings Como, Italy, 2001.
Gausemeier, J., Frank, U., Donoth, J., Kahl, S., "Specification technique for the description of self-optimizing mechatronic systems", Research in Engineering Design, 20, 2009, pp. 201-223.
Herberg, A., Lindemann, U., "Supporting system architecture assessment by two-level functional modeling: Process oriented analysis and comparison of large-scale systems", IEEE International Systems Conference, 2012, pp. 1-8.
Kernschmidt, K., Bayrak, G., Vogel-Heuser, B., "A port-based approach for modeling the structure of mechatronic modules", 14th International Dependency and Structure Modelling Conference, Kyoto, Japan, 2012, pp. 111-123.
Modarres, M., "Functional modeling of complex systems with applications", Annual Reliability and Maintainability Symposium, IEEE 1999, pp. 418-425.
Modarres, M., Cheon, S. W., "Function-centered modeling of engineering systems using the goal tree–success tree technique and functional primitives", Reliability Engineering & System Safety, 64, 1999, pp. 181-200.
Ponn, J., Lindemann, U., "Konzeptentwicklung und gestaltung technischer produkte: Systematisch von anforderungen zu konzepten und gestaltlösungen", Springer, 2011.
Rapp, T., "Produktstrukturierung: Komplexitätsmanagement durch modulare produktstrukturen und-plattformen", BoD-Books on Demand, 2010.
Rodenacker, W., "Methodisches konstruieren", Springer-Verlag, Berlin, 1971.
Roth, M., Kasperek, D., Lindemann, U., "Functional analysis and modeling of complex, evolutionary grown, mechatronic products", working paper, 2013.
Rui-Qin, L., Hui-Jun, Z., "A new symbolic representation method to support conceptual design of mechatronic system", The International Journal of Advanced Manufacturing Technology, 25, 2004, pp. 619-627.
Soley-technology, Available at: <http://www.soley-technology.com/>, 2014.
Sørensen, M. U., "Application of functional modelling in the design of industrial control systems", Reliability Engineering & System Safety, 64, 1999, 301-315.
Stark, R., Beier, G., Wöhler, T., Figge, A., "Cross-Domain Dependency Modelling – How to achieve consistent System Models with Tool Support", 7th European conference on systems engineering research, Eusec 2010, Stockholm, 2010.
Umeda, Y., Takeda, H., Tomiyama, T., Yoshikawa, H., "Function, behaviour, and structure", Applications of artificial intelligence in engineering, 1, 1990, pp. 177-194.
Van Beek, T. J., Erden, M. S., Tomiyama, T., "Modular design of mechatronic systems with function modeling", Mechatronics, 20, 2010.
Van Beek, T. J., Tomiyama, T., "Connecting views in mechatronic systems design, a function modeling approach", IEEE/ASME International Conference on Mechtronic and Embedded Systems and Application, Beijing, China, 2008a, pp. 164-169.
Van Beek, T. J., Tomiyama, T., "Requirements for complex systems modeling", The 18th CIRP Design conference—design synthesis, 2008b.
VDI, "Vdi guideline 2206: Design methodology for mechatronic systems", Beuth Verlag, Berlin, 2004.
Weilkiens, T., "Systems engineering with sysml/uml: Modeling, analysis, design", Heidelberg, 2011.
Weyrich, M., Klein, P., Laurowski, M., Wang, Y., "A function-oriented approach for a mechatronic modularization of a sensor-guided manufacturing system", 2011, pp. 12-16.
Wölkl, S., Shea, K., "A computational product model for conceptual design using SysML", Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference , San Diego, California, 2009.

Dipl.-Ing. Mr. Daniel Kasperek
Technical University Munich, Department of Product Development
Boltzmannstrasse 15, 85748 Garching, Germany
Email: kasperek@pe.mw.tum.de