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

DESIGN 2010

A VIEW-BASED MODELING APPROACH FOR REPRESENTING MULTIDISCIPLINARY FUNCTIONS IN PDM SYSTEMS

M. Eigner, K.G. Faißt, T. Hollerith and F. M. Nem

Keywords: view-based modeling, mechatronical function modeling, product lifecycle management

1. Introduction

Engineering enterprises today have to cope with various challenges. Amongst others, the pressure of today's competitive and globalized business contexts and the increasing complexity of nowadays multidisciplinary products force enterprises to reduce their product development time and cost by simultaneously accounting for product and environmental legislations becoming more and more stricter, and by guaranteeing their customers permanent product innovations, high product quality and tailored product choices. In case of mechatronical product development, a survey made by the AberdeenGroup Inc. shows that 68% of mechatronical product manufacturers face the problem of the synchronization of mechanical and electrical design representations. Furthermore, 36% face the problem caused by the existence of heterogeneous discipline-specific data management tools, and 28% face the problem caused by the heterogeneity of discipline-specific design processes [Jackson 2006]. Further, another survey showed that the problem of implementing an integrated product development solution for all disciplines involved in mechatronical product development (28%) and the ability of understanding the impact a change will have across disciplines (18%) are also crucial [Boucher 2008], even though considering that the most significant under the top six challenges during mechatronical product development are: the difficulty to find and to hire experienced system engineers respectively the lack of cross-functional knowledge (50%), and the early identification of system level problems (45%). Therefore, beyond the provision of new product development methodologies for multidisciplinary products (e.g. the design methodology for mechatronical systems also known as VDI 2206), mature, tailored and user-accepted supporting engineering and management tools have to be supplied in order to ensure the successful development of qualitative and reliable mechatronical products. From a PLM (Product Lifecycle Management) perspective, the most troublesome gaps which have to be bridged in this regard are the integration, federation, structuring, synchronization and management of disparate and complex engineering partner-specific or discipline-specific product data and related engineering processes, as well as the management of their configurations and their high number of variants.

In previous works [Mogo Nem 2008, Eigner 2009], the concept of Engineering Networks (EN) has been introduced. It represents a holistic and integrated modeling approach for product and process related data in engineering and addresses, amongst others, the above-mentioned challenges of multidisciplinary PLM. This paper focuses on the view-based modeling approach used inside EN to model, structure and manage multidisciplinary product information along the entire product lifecycle. Also, a prototypical implementation of this concept using the example of multidisciplinary product functions is provided.

Therefore, the following chapters of this paper are organized as follows: section 2 addresses related works dealing with multidisciplinary product data management as well as view-based modeling. In section 3, multidisciplinary function modeling is presented as well as the intended application of a view-based modeling approach to map product functions into a data management system. The concept for the proposed view-based modeling approach is outlined in section 4 and prototypically implemented in section 5, using a low cost PDM system. In section 6, a conclusion is given and further works will be addressed.

2. State of the Art and Related Works

Nowadays, for the management of product related data during product development enterprises use tailored and different Product Data Management (PDM) systems for the discipline Mechanics and Electronics/Electrics as well as Software Configuration Management (SCM) systems in case of Software Engineering. Additionally, companies use Workflow Management (WfM) systems for the deployment of the associated engineering processes. For the computerized representation and management of products and engineering processes, product data and process models are used accordingly. In order to span the whole product lifecycle, the new paradigm of Product Lifecycle Management (PLM) has been introduced. Several mature and even standardized product data and process models are nowadays available. One of the famous standardized object-oriented product data models is the ISO Norm 10303 series also known as STEP (STandard for the Exchange of Product data). Regarding mechatronical product development, the differences in partner- or discipline-specific product data models involved make it difficult to develop a semantic global product data model for Mechatronics. Many research activities have proposed to integrate the standardized discipline-specific product data models defined in STEP to a global integrated product data model using object-oriented aggregation [MechaSTEP 2001]. Doing so, the discipline-specific partial product data models are aggregated to the mechatronical product data model by usage of view elements. Although such an approach offers a good way to bring together discipline-specific product data in order to form a mechatronical product, the usage of aggregation to link together the disparate discipline-specific partial product models has some disadvantages. First, the semantics associated with each discipline-specific view is missing, making it difficult to semantically relate information to separate product views. Second, adequate algorithms need to be implemented in order to extract from the resulting information cloud the relevant data for a specific user. However, solutions for these disadvantages can be found through the provision of a modeling concept for product data based on the consideration of the different views of the involved users. A general view-centered modeling paradigm has been proposed by Nassar for modeling complex software systems accessed by different users with different requirements and information needs [Nassar 2003]. In the further, this approach is followed up and extended to support multidisciplinary product modeling, e.g. by allowing the assignment of multiple views of an object to an user by using a "viewpoint", or by allowing a view to reference a subset of information of the object containing it.

3. Multidisciplinary Function Modeling

Systems Engineering has been established today as a bridge between Requirement Analysis and System Design phases during product development. In case of a mechatronical product, it enables the development of interdisciplinary solution concepts based on required product functionalities. Doing so, multidisciplinary function modeling has become a crucial engineering task aiming at identifying the main functions of a product at a multidisciplinary level, which are used later on as input for the subsequent engineering discipline-specific development of solution elements. Functions abstract from the realization of product functionalities and could therefore be realized using different discipline-specific solution elements. This variability in the realization of a function or a group of functions influences the resulting number of possible product variants to be offered to the customers. Functions are structured and interrelated to form a function model. This represents another view on the product being engineered in contrast to the designed product structure, since customer relevant aspects are represented in opposite to technical aspects in case of the latter. The functional structure (based on customer needs) also enables the linkage of product requirements and product structure [Lindemann

2005]. In this regard, Lohse has proposed to link assemblies and single parts of a complex product to their corresponding functions [Lohse 2001]. Since such an assignment can bear some difficulties in that case where several assemblies of a product fulfill multiple functions, he proposed to benchmark product assemblies and the usage of strict allocation guidelines. According to these guidelines, it is possible to match a requirement to its corresponding product function.

3.1 An Approach for Multidisciplinary Function Modeling

In this section, it is outlined how functions can be used as "glue" between the product requirements and the subsequent product structure in case of Mechatronics. The proposed approach consists of three subsequent steps as depicted in the Figure 1 below.

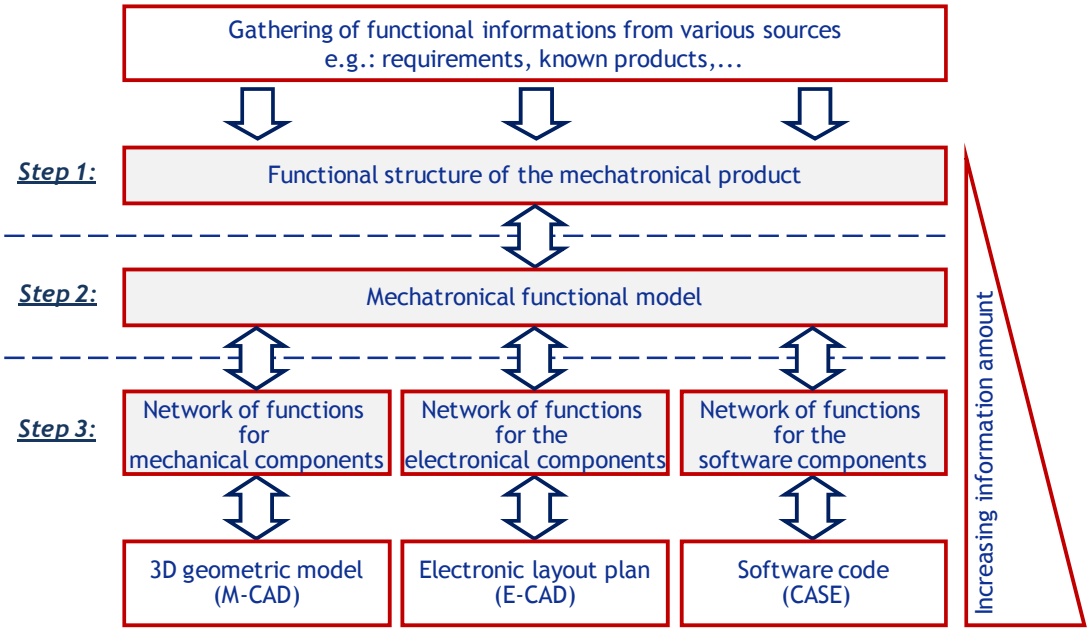

Figure 1. A multidisciplinary function modeling approach

The three steps approach for multidisciplinary function modeling can roughly be compared with the widespread paradigm of Model Driven Architecture (MDA) for software systems launched by the OMG (Object Management Group). In the MDA paradigm, a software system is developed outgoing from a Platform Independent Model (PIM) which is successively transformed and enriched until a set of executable artefacts denoted by a Platform Specific Model (PSM) representing the desired Software solution is finally generated.

In the first step, our approach deals with the derivation of multidisciplinary product functions based on product requirements. During this first step, not only the targeted functions are considered but also the extraction of customer, engineering, economical as well as environmental properties (e.g. the maximal velocity, weight or CO2 emission of a car) significant for the control of the whole multidisciplinary product development and their assignments to functions too. Actually, such product properties are only captured in listings while a formal structure with the dependencies and relations between these various properties is still missing [Faisst 2007]. In the next subsequent step, the identified multidisciplinary functions are structured in order to form a function model. This function model is comparable with a PIM in terms of the MDA paradigm since the functions it is including are still independent from any concrete information regarding its realization in a specific discipline. During the last step, some functions of the multidisciplinary function model are assessed and assigned to the involved disciplines for their realization. The resulting discipline-specific subsets of the overall multidisciplinary function model are used for subsequent discipline-specific engineering tasks. They are further specified and enriched with discipline-specific information regarding their realization and can therefore be compared to the targeted PSM in terms of the MDA paradigm.

3.2 Necessity of a View-Based Representation of Multidisciplinary Functions

Even though Systems Engineering has been well established during mechatronical product development, there is still no standard prescribing how a function model for multidisciplinary products should be represented and mapped into a data management system. However, nowadays several tools and models are available for modeling and even simulating multidisciplinary functions. Examples for widespread tools are MATLAB/Simulink or Modelica. They offer a graphical user interface for the modeling of a system using a library of functional elements.
For the proposed approach of function modeling, to simplify matters we follow only the common elements depicted in Figure 2 as being parts of a multidisciplinary function model.

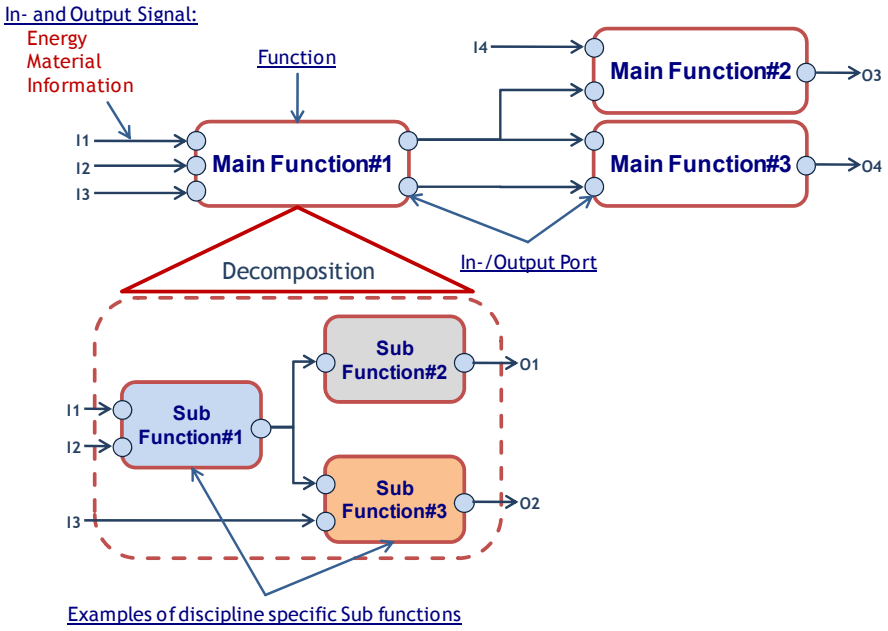

Figure 2. Constituents of a multidisciplinary function model

Now, it is important to consider how subsets of a multidisciplinary function model are built and assigned to specific disciplines. A very important aspect in this regard is the fact that a single multidisciplinary function can be realized using various solution principles and therefore can be assigned to multiple disciplines. Moreover, a function previously assigned to a specific engineering discipline for its realization could, eventually during its refinement, appear to contain multidisciplinary sub-functions. Another interesting aspect is the sharing of a function among multiple disciplines. All these considerations raise interesting requirements for the representation and management of multidisciplinary functions, and multidisciplinary product data in general. A simple approach to address these requirements could be to make several copies of an initial multidisciplinary function model which are thereafter adapted through the deletion of not relevant functions in order to build desired discipline-specific subsets. Thereupon, user access privileges to these subsets can be defined. Some drawbacks of this approach are, apart from the associated data redundancy and the overhead to keep these separate copies of a function model up-to-date, the loss of the required modeling correctness since a multidisciplinary function is a discrete element with different interpretations and structures depending on the discipline realizing it. This motivates the consideration of a view-based modeling and management paradigm as described in the following section.

4. The Proposed View-Based Modeling Approach Using Engineering Objects

The fundament of the modeling approach proposed here is the extension of object-oriented product data modeling approaches with the new concepts of *View* and *Viewpoint* as part of a new type of objects denoted *Engineering Objects (EO)*. Considering the two modeling abstraction levels *Model*

and *Instance*, an EO represents at *Instance* level an (multidisciplinary) object and is defined at *Model* level by an appropriate EO-Type (similar to a *Class* in the object-oriented modeling paradigm).

4.1 The Concept of Engineering Objects

An EO can be used to represent any kind of information (similar to an object in the object-oriented modeling paradigm). In the context of product engineering, an EO can represent for example a complete product (e.g. a car), a single part, an assembly of parts, or even a product function. The focus lies on the description and definition of an EO by its properties and their structure given from the view of the involved persons. Therefore, an EO must always be linked to the people involved in developing or manufacturing a product. The involved persons are called *Observers*, or, keeping in mind that they are active persons in the process, *Actors*. These Observers may be individuals or groups of persons with similar range of tasks, interests and/or education. Due to the different views of Observers, rather different product properties are of interest for them [Faisst 2007, Faisst 2004]. This concept of Engineering Objects is used as fundament for the intended view-based modeling and access mechanism for multidisciplinary product information due to its inherent observer-dependent structuring of information. For its formalization, the concepts of *View* and *Viewpoint* have been used [Mogo Nem 2008, Eigner 2009].

A View of an EO is used for integrating, structuring and abstracting the information making up an EO. A View can encapsulate information relevant for a specific partner, engineering discipline, application field in an engineering discipline, or even an activity inside an engineering process manipulating an EO. In case of multidisciplinary product engineering, a View can represent the information available in a partner- or discipline-specific data management system, or shared by multiple partners or disciplines at a specific product lifecycle stage. A common concept of View is widespread in the database technology as an important mechanism for providing logical data independence, data hiding and is also a mean for data integration. However, a View is assimilated there to a kind of filter on a global model making the definition of dependencies between them difficult. In opposite, we consider a View of an EO as integrated part of it which is also defined during the modeling of the corresponding EO-Type. Once an EO of a specific EO-Type is instantiated, memory is allocated for representing each View defined in its EO-Type. Therefore, an EO can be thought of as a kind of intelligent object having enough information about itself and being able to behave differently depending on the Observer accessing it. A View of an EO can also reference some EO's properties or can define its own in order to build a view-specific subset of the overall EO's properties. Consequently, the total amount of properties for an EO is the union of its view-specific properties and those directly assigned to it.

A Viewpoint, however, is the combination of multiple Views of an EO-Type which can be assigned to a user or a group of users in order to regulate the access to its instances (EO). A user having a specific role has only access to those Views of an EO available in the corresponding Viewpoint definitions. This view-based access mechanism is advantageous compared to traditional access mechanisms where access privileges to an object are granted to specific users through their roles for the complete object including all its properties and methods. Doing so, the role-specific access to subsets of object properties and methods in general unfortunately is solved during the development of Graphical User Interface (GUI). Through the provision of a view-based access mechanism, the information available for an EO can be structured so that a user having a specific role, to which a specific Viewpoint is assigned, can only access the information available in the Views defined for that Viewpoint. Additional use of the concepts of View and Viewpoint is done in the holistic concept of Engineering Networks to support a role-specific generation of adequate GUI for manipulating EO.

4.2 Abstraction of the Metamodel Realizing the Concept of Engineering Objects

The first step towards the realization of the view-based modeling concept using Engineering Objects as part of the holistic concept of Engineering Networks (EN) has been the definition of a suitable metamodel. Doing so, the standardized metamodeling language MOF (Meta Object Facility) developed by the OMG has been chosen. Using the MOF language allows to describe the metamodel in an UML (Unified Modeling Language) similar notation and to leverage already existing standardized metamodels like MOF itself or UML. The metamodel for EN is structured into four main

packages. The core of this metamodel is the package *ENCore* which contains the definition of Engineering Objects. Describing the whole content of this package would go beyond the scope of this paper. So, in this paper the focus is only on fundamental elements used for defining Engineering Objects as shown in Figure 3.

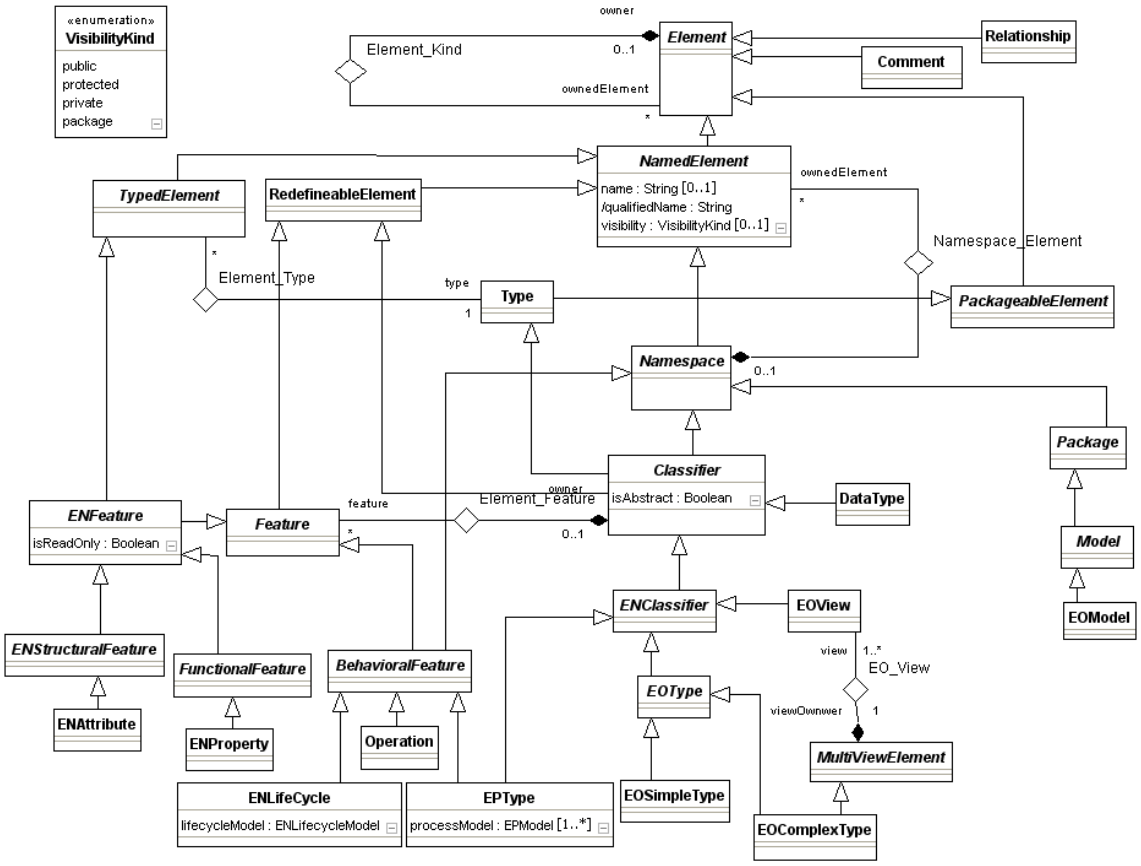

Figure 3. Abstraction of the content of the Package ENCore in Engineering Networks

The element of special interest is the metaclass *ENClassifier* from which the two metaclasses *EOType* and *EOView* are derived. *EOType* is used for defining at modeling level an EO-Type describing several EO, while *EOView* defines the associated View definitions. In order to support the traditional object-oriented modeling concept of *Class*, as specified for example in UML, we further sub-classify *EOType* in *EOSimpleType* and *EOComplexType*. The difference between them is the support of the concept of View by the latter, whereas *EOSimpleType* doesn't support View definition and is therefore the same as traditional widespread concept of *Class*. For an *EOComplexType*, at least one View definition should be provided. Relationships between several *EOType* are also supported in the metamodel but are specified in a separate package. However, in order to provide the intended view-based modeling and user access management concept, it is only allowed to define a relationship (simple directed or undirected association, aggregation or composition) among several *EOComplexType*, among similar View definitions in several *EOComplexType*, among several *EOSimpleType* or between an *EOView* and several *EOSimpleType*.

5. Prototypical Implementation and Validation of the Proposed Concepts

In order to validate our view-based modeling approach, an implementation of a simplified multidisciplinary function model of the driving system of a RC car as shown in Figure 4 has been used using the free available PLM Software Aras Innovator.

Starting point for the development of such a driving system is the analysis of its requirements. Those are for example the expected operating power (between 10V and 12 V), the expected operating environmental temperature (between 10 and +40°C), the expected maximal torque or the braking

power of the car. Based on that, multidisciplinary functions and product properties can be identified. We distinguish between multidisciplinary functions such as *Accelerate car* or *Create braking power* and discipline-specific functions such as *Regulate torque* or *Create torque* assigned to Software and Electrics/Electronics respectively. Multidisciplinary functions are per default accessible by every user involved whereby only the information available within the respective discipline-specific views is displayed. This enables for example to manage a multidisciplinary function for which several discipline-specific realizations are available using a single object (here an EO) in a PDM system.

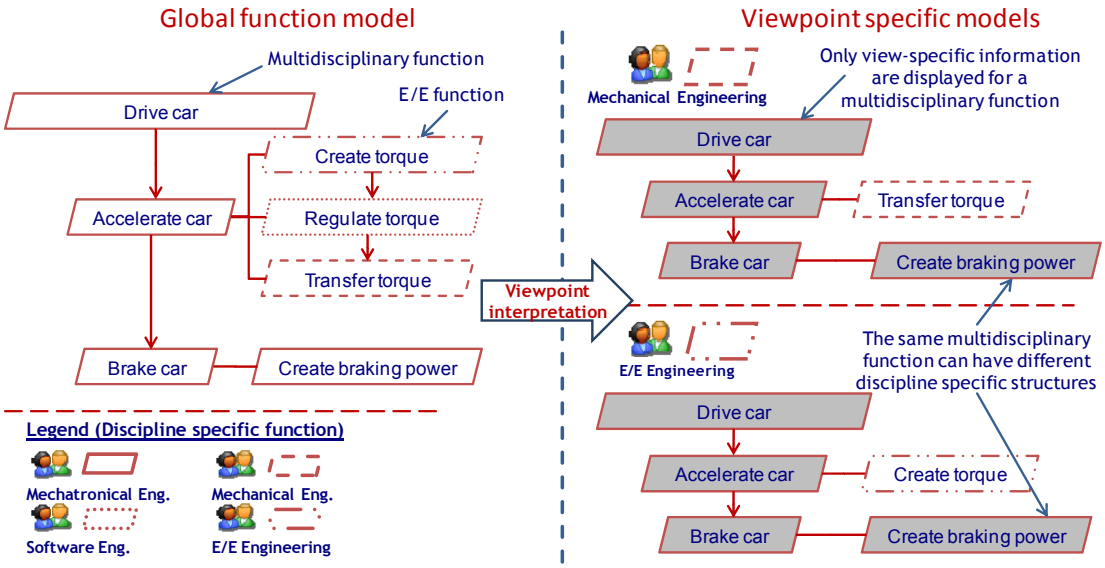

Figure 4. A simplified multidisciplinary function model for a RC car

5.1 Basic Object-Oriented Modeling Concepts Provided in Aras Innovator

Aras Innovator is based on a framework (*Aras Innovator Core*) providing powerful object-oriented modeling concepts for fast implementation of business solutions. Figure 5 gives an overview of the basic modeling artefacts provided by the Aras Innovator framework. The basic artefact is named *ItemType* which is comparable to a *Class* in terms of object-oriented modeling.

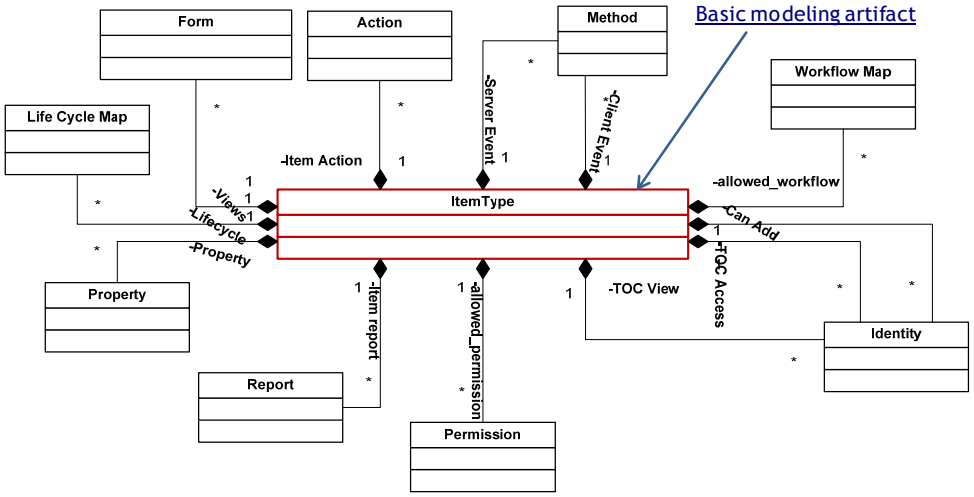

Figure 5. Basic modeling artifacts in Aras Innovator

Any business object managed in Aras Innovator is called an *Item* which is an instance of an *ItemType*. For each *ItemType*, amongst others, *Property* for representing its attributes, *Actions* and *Methods* implementing required behaviours of instance of that *ItemType*, as well as *Permission* for defining

user access privileges to its instances can be specified. User access rights are not directly assigned to logon users in Aras Innovator. Instead, a role-based access control mechanism is used. Doing so, a logon user is assigned to one or more *Identity* which is similar to a role or role group. Access privileges are granted only to *Identity* and not directly to a logon user. Upon creation and saving of an *ItemType*, a default *Form* is automatically generated by the Aras Innovator framework in order to access and manipulate its instances. Several other *Forms* can later on be defined and customized and assigned to specific user identities. The Aras Innovator framework also supports the definition of lifecycle diagrams (*Life cycle Map*) capturing the dynamical evolvement of Items, as well as the specification of several business processes (*Workflow Map*) operating on them.

5.2 Implementation of the Required View-Based Modeling and Management Concept

Since the Aras Innovator framework does not directly support a view-based modeling approach, a workaround has been necessary. The prototypical implementation has been started by defining a simplified organizational structure consisting of several logon users and their assignments to a hierarchical structure of identities with different privileges as given in Figure 6.

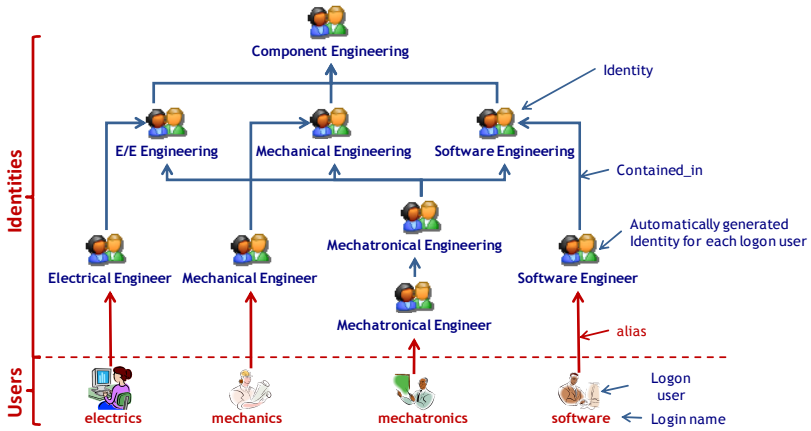

Figure 6. Simplified organizational structure for mechatronical product development

Thereafter, several ItemType definitions are necessary in order to define a suitable model for simulating the expected view-based modeling and access concept. In this context, the most important modeling element is the ItemType *Function* which represents function in Aras Innovator. Every function managed in Aras Innovator is therefore an instance of *Function* and has per default three discipline-specific Views which are represented by using the ItemTypes *Function_View_ME*, *Function_View_EE* and *Function_View_S* respectively for Mechanics, Electrics/Electronics and Software. These Views are related to the ItemType *Function* using several relationships. Providing solely this ItemType for representing both multidisciplinary and discipline-specific functions brings some advantages. In early mechatronical function modeling steps, the functions created cannot always directly be assigned to a specific discipline and are initially considered being multidisciplinary, whereby discipline-specific users accessing them have only access to their corresponding Views. Later on, such a function can be, if necessary, dynamically assigned to only one specific discipline and thereupon is only visible for associated discipline-specific users. Further, the ItemType *Function_ENProperty* has been created to allow the definition and assignment of design related product properties to functions or their Views. Doing so, it is possible to subdivide the overall properties related to a function into discipline-specific subsets by directly assigning them to the corresponding View (in this case, these properties are only accessible by users having access to that View) or by assigning them first to the function itself and relating them afterwards to the desired Views (this is helpful to share properties of a function across several disciplines). Still missing so far is the support of interrelating functions in order to build function structures. In this regard, and with respect to the intended view-based modeling concept, several types of relationships have been provided for interrelating functions. The first type allows the interrelation of a function with other (sub-) functions independently from their Views. In such a case, the semantics of the relationship is

valid across the discipline borders. This type of relationship should be used to decompose a multidisciplinary function into its multidisciplinary constituents. The remaining types of relationships support only the discipline-specific interrelations of functions through their discipline-specific Views. For example, in order to further specify a function *F1* assigned to mechanical engineering into its discipline-specific sub-functions *F2* and *F3*, relationships should be defined from the mechanical view of *F1* to those of *F2* and *F3*.

By now, the required model and the organizational structure have been defined. Now, the provision of a view-based access concept should be considered. In Aras Innovator, only the traditional role-based access control of users to items is supported. This has to be bypassed using customized software code in order to allow users to access only the Views of a function which have been assigned to their respective Viewpoints. So, customized software code has been written to create per default automatically for each new created function three discipline-specific Views related to it, regardless of its assignment to a specific discipline. Then, depending on the identity of the creator, a discipline-specific assignment is performed automatically. E.g. according to Figure 4 and 6: if the logon user *mechanics* creates a new function, this function is automatically assigned to Mechanics and only accessible by users with similar identity. However, when a new function is created by the user *mechatronics*, it is first considered as mechatronical and its discipline-specific Views as well as its multidisciplinary structure are accessible by any discipline-specific users until it has been assigned to a discipline later on, if necessary.

5.3 Validation of the Implementation

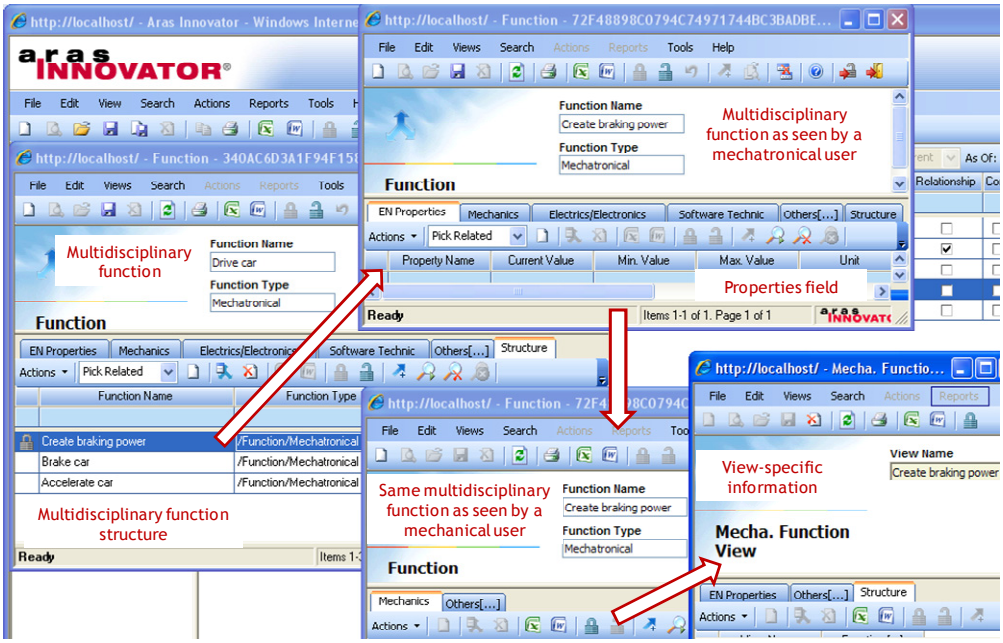

Figure 7. Screenshots of the implemented solution and a view-based access to information

In order to validate the implementation, the function model given in Figure 4 has been mapped into the developed model inside the Aras Innovator as shown in Figure 7.

6. Conclusion and Outlook

This paper presents a view-based modeling approach for modeling, structuring and accessing multidisciplinary product information using the example of multidisciplinary product functions. The underlying modeling approach is based on the concept of Engineering Objects which is part of a holistic approach called Engineering Networks currently under development in a research project. A prototypical implementation of the concepts proposed has also been done using the free available PDM Software Aras Innovator. Only the mapping and management of function structures is supported so far. In further work, the current implementation will be extended in order to support additional

information about a function like, for example, the associated in- and output signals. Further, it is intended to provide identical implementation for modeling, structuring and managing multi-disciplinary requirements and product structures as well as to support multidisciplinary and discipline-specific interrelating of requirements, functions and product structures for a holistic view-based multidisciplinary product lifecycle management.

Acknowledgments

This paper is based on work achieved in a current research project funded by the German Federal Ministry of Economics and Technology (BMWI). We would like to thank the BMWI for supporting us.

References

Boucher, M., Houlihan, D., "System Design: New Product Development for Mechatronics", Aberdeen Group, 2008, http://www.aberdeen.com/.

Eigner, M., Mogo Nem, F. "Conceptual modeling and generator framework for multidisciplinary and collaborative Product Lifecycle Management", Proceedings of the 13th International Conference on Computer Supported Cooperative Work in Design, Santiago/Chile, 2009, pp. 590-595.

Jackson, C, "The Mechatronics System Design Benchmark Report - Coordinating Engineering Disciplines", AberdeenGroup, 2006, http://www.aberdeen.com/.

Faisst, K., Dankwort C., "Aesthetics in a Formalised Reverse Design Process, Proceedings of the 8th International Design Conference - DESIGN 2004, Dubrovnik/Croatia, 2004, Vol. 1, pp. 197-204.

Faisst, K., Dankwort C., "New Extended Concept for the Usage of Engineering Objects and Product Properties in the Virtual Product Generation Process", Proceedings of 16th International Conference on Engineering Design: ICED 07 - Knowledge, Innovation and Sustainability, Paris, 2007, pp. 689-690.

Lindemann, U., Ponn, J.: "Methodische Entwicklung technischer Produkte, 2. bearbeitete Auflage, Springer-Verlag Berlin Heidelberg, 2005.

Lohse, A., "Auftragsmanagement von komplexen Produkten in agilen Unternehmensstrukturen, PhD Thesis, Fakultät für Maschinenbau und Elektrotechnik, Technische Universität Carolo-Wilhelmina Braunschweig, 2001.

MechaSTEP Project, "STEP data model for simulation data of mechatronic systems", DIN PAS 1013, Beuth-Verlag, 2001.

Mogo Nem, F., Weidlich, R., Eigner, M., "Engineering Networks: A concept for the coequal modeling of data and processes in the product engineering", Proceedings of the 10th International Design Conference - DESIGN 2008, Dubrovnik/Croatia, 2008, pp. 849-856.

Nassar, M., Coulette, B., Crégut, X., Ebersold, S, "Towards a view based Unified Modeling Language", Proceedings of the 5th International Conference on Enterprise Information Systems, 2003, pp. 257-265.

Dipl. -Inf. Fabrice Mogo Nem
Research associate and PhD candidate
University of Kaiserslautern, Institute for Virtual Product Engineering (VPE)
P.O. Box 3049, 67653 Kaiserslautern, Germany
Telephone: +49 631 205 2312
Email: mogo_nem@mv.uni-kl.de
URL: http://vpe.mv.uni-kl.de