INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

ENGINEERING NETWORKS: A CONCEPT FOR THE COEQUAL MODELING OF DATA AND PROCESSES IN PRODUCT ENGINEERING

F. Mogo Nem, R. Weidlich and M. Eigner

Keywords: engineering object, engineering process, object orientation, model driven architecture

1. Introduction and motivation

Enterprises today are in constant change and have to deal with new arising challenges. Amongst other factors, the introduction of mechatronic products and the globalisation result in closer and more direct collaboration between enterprises, their suppliers and their customers. Whereas the rising demand for product innovation, product reliability and product liability have caused new challenges in the product- and process management. The term product in this paper is used as a synonym for part or assembly of parts. The information about a product throughout its complete lifecycle, from the early phases up to recycling, often is distributed across a global network of data handling systems, supporting the different lifecycle phases and the different engineering disciplines (see Figure 1).

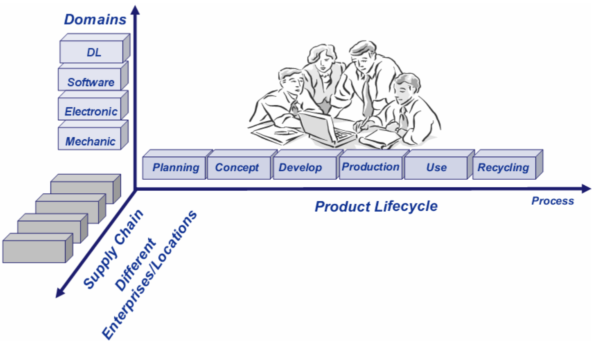

Figure 1. Dimensions influencing enterprises today

The capability of enterprises and their IT-systems to support the integration and federation of the distributed product data and their related processes, as well as the administration of the high number of product variants have already been recognized as driving factors for success, survival and competitiveness. Although Product Data Management (PDM) and Product Lifecycle Management (PLM) systems offer first answers to these problems, their use in the practice remains still restricted. This is mainly due to the high costs caused by the purchasing, customizing, introduction and maintenance of such systems. Therefore the problems mentioned above are solved with unconventional means or are often manually bridged, as many, mainly small and medium-sized, enterprises, can not afford PDM/PLM-systems.

With the concept of Engineering Networks (EN) described in this paper, we want to make our contribution to a solution of these problems. The concept of Engineering Networks describes a meta-model, which improves the modeling of product data and engineering process models in enterprises. It provides constructs for the efficient modeling of the current needs for integrated and federated product data and engineering process models and supports their mapping into data management systems. The supply of product and process information, with the consideration of the different information needs of users in a complex network of customers and suppliers, as well as information reuse is supported on the meta-model level. This is done inter alia by means of an extended view concept.

The implementation of the concept of Engineering Networks provides a powerful and innovative modeling platform for product data and engineering process models. This platform can be used to efficiently model, maintain and extend product information, including its associated engineering processes throughout its lifecycle. Further, providing a neutral and accessible meta-model enables the usage of the widespread Model Driven Architecture (MDA) approach to support the automated creation of executable codes from a given model for a specific software platform. The use of MDA is a key to the development of a new generation of PDM and PLM systems, which is easier to implement, to customize and is more independent of proprietary solutions. This new generation of PDM/PLM-systems clears the way for a more efficient product data and engineering process management in small and medium-sized enterprises.

The remainder of this paper is organized as follows: Section 2 introduces the ideas behind the Engineering Networks concept; Section 3 discusses some approaches to realise Engineering Networks; Section 4 shows one example of an application field for Engineering Networks and the last Section gives a conclusion and perspectives for further work.

2. The Concept of Engineering Networks

The increasing importance of mechatronics, the globalisation throughout the value creation chain and the necessity to continuously support the different product lifecycle phases are the three dimensions influencing the enterprises today. Globalisation has led to the dislocation of enterprise units over the whole world and to the increasing need for collaborative work between enterprises, their suppliers and their customers. The interaction of mechanical, electronic and information systems in mechatronic products offer great potentials for product innovation. The partitioning of the value creation chain in different lifecycle phases and the application of the methods of virtual product development aim at reducing the product development times and increasing the product quality. However, a holistic IT-support of these three dimensions represents a challenge for many enterprises. This is especially true for the development of mechatronic products, where the creation of a global view on the distributed product information remains a big challenge. With Engineering Networks, we intend to address all of these problems. An Engineering Network is a meta-model and enables the modeling of federated and integrated product data and engineering process models. It is based on the concepts of *Engineering Object (EO)* and *Engineering Process (EP).* Engineering Objects are used to model the product relevant information and support its user dependent presentation. Engineering Processes are used to model the relevant engineering processes associated to the Engineering Objects. This section introduces Engineering Object and Engineering Process as components of Engineering Networks.

2.1 The concept of Engineering Object

In order to focus on virtual product development, all the relevant information of a product throughout its lifecycle must be mapped to a digital model using suitable product data models. The resulting digital product is managed today by PDM/PLM-systems. Particularly for mechatronic products, the differences in the data models of the three involved engineering disciplines (software, electric/electronic and mechanic) make it difficult to develop a global product data model. Instead, in practice the different partial models of these disciplines are modelled separately, following the object-oriented paradigm and are linked together using the principles of aggregation or composition. Many partial models are already available in practice and some of them are even standardised and can be used as neutral formats for exchanging product data. Examples of such standardised partial models are defined in the different parts of the ISO Norm 10303 also known as STEP (STandard for the Exchange of Product Data). With the concept of Engineering Object, we don't want to define a new product data model to replace existing and established ones. Rather, we define a new meta-model which enables a better definition of product data models. The concept of Engineering Object is not a new one; it has already been defined and refined in previous works at the institute for Virtual Product Engineering (VPE) of the University of Kaiserslautern. An Engineering Object represents an object within a given state in the product development process [Faisst 2007]. It can be used for example to represent a single requirement object, a product component (a single part art or even a complete assembly of parts like a car) or a marketing plan. One of the most important features of an Engineering Object, which distinguishes it from common objects in the object-oriented world, is its *virtuality*. An Engineering Object has no relevance in reality when it is considered as a *stand-alone* object without any, so called, observer. That means, the properties of an Engineering Object depend on the viewpoint of the observer and the activated view. The term view is used to represent an abstraction of a model and the term viewpoint represents the sight that has an observer of this model [Hair 2004].

A view represents a subset of properties that are relevant for a given application field. In Figure 2 the *Requirement View* is used to group all the properties which are relevant for requirement management, the *Marketing View* is used to group all the properties which are relevant for marketing and the *Design View* is used to group all the properties which are relevant for design.

A viewpoint is the combination of multiple views of a model and can be assigned to an observer or an observer group. The *Manager viewpoint* in Figure 2 is the combination of the *Marketing*- and the *Requirement View*. The observers or observer groups the *Manager viewpoint* is assigned to only have access to the *Marketing*- and *Requirement View*.

In the example only three views and two viewpoints are shown. Depending on use cases additional views and viewpoints like the *Manufacturing View*, the *Recycling View* or the *Stylist Viewpoint* are possible.

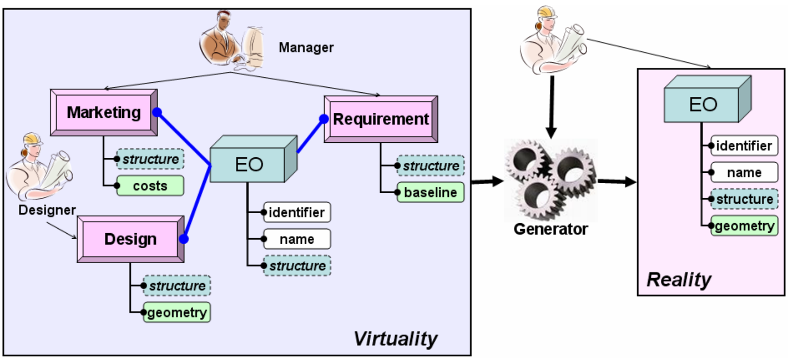

Figure 2. Virtuality of Engineering Object

In order to avoid information redundancy by the definition of views, we subdivide the set of properties of Engineering Object in three disjoint subsets. The first subset called *common subset* consists of properties which are shared by all the views. The values of these properties are the same in all views. For example the properties *identifier* and *name* of the Engineering Object in Figure 2 are the same in the three views. The second subset called *the variable subset* consists of properties which are also available in all the views; however their concrete realisation can differ from one view to another. For example, each of the three available views in Figure 2 has an associated property called *structure*. It represents a view specific structure of that Engineering Object. But the structure of the *Requirement View* differs from that of the *Design View*. The first represents a structure of requirement objects and the second represents a structure of assemblies and parts. The last subset called *specific subset* is built up by properties which are defined in the different views and only available therein. For example, the *Design View* in Figure 2 defines the property *geometry* and the *Marketing View* the property *costs*. Each of these properties belongs only to the view defining it. During runtime, when a specific observer

viewpoint is specified and one of the available views for that viewpoint is activated, all the information available for that view is brought together and presented to the observer.
The virtuality of Engineering Object provides several improvements in modeling complex product like mechatronic product. It enables the building of a unique model accessible by different observers with various needs and also supports the management of observer access rights.
Another important feature of Engineering Object is the distribution of their properties over complex networks of data handling systems. That means, the properties of an Engineering Object can be distributed over many systems throughout the world. These properties have to be brought together at runtime and presented to the observer.
For the usage of Engineering Object in the context of Engineering Networks as described above, they will be extended by three other important features: its *graphical presentations*, its *service interfaces* and its *engineering processes*. The graphical presentations are used to enable the interfacing of an EO with their observers. They are used to present an EO in a Graphical User Interface (GUI) environment. So the observers can use these interfaces to interact with or to manipulate an EO. The linkage of EO properties to graphical components is very important for implementing application systems which are easy to customize. It also enables the automated change of the graphical presentations when an EO property is changed. The service interfaces are used to access an EO by other application systems. They are specified at design time and are used to build up the complete interfaces of the implementing application system. So, a modification of EO interfaces automatically modifies the interfaces of the application system implementing it. This reduces the costs and the implementation effort needed to keep interfaces of application systems up-to-date. The definition of service interfaces is based on the web services technology. The linkage of an EO to its engineering processes is discussed in the next section.

2.2 The concept of Engineering Process

The Workflow Management Coalition defines the term business process as: "*A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, …*" [WfMC 1999]. As defined above, the aim of a business process is the realisation of a business objective. Examples of business processes are patient registration in hospitals or airplane construction. In [Bitzer 2008], six different enterprise process levels are distinguished (See Figure 3).

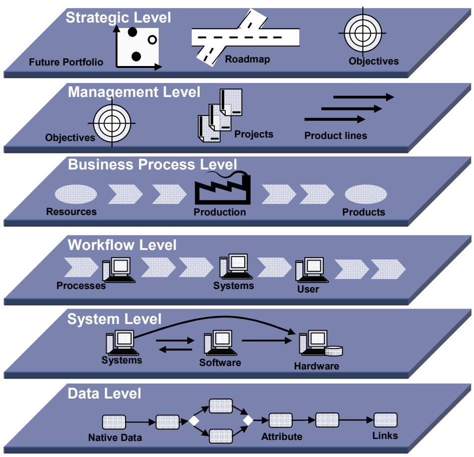

Figure 3. Process levels [Bitzer 2008]

On the Strategic and the Management level, visionary and business strategic targets are defined and transferred into operative targets on the Business Process level. The Business Process level represents the activities aiming at value creation in enterprises. The Workflow level connects the Business Process level with the System level, which includes information about used systems, software and

hardware. Product data are stored on the last level, the Data level. The term process in the context of Engineering Networks is restricted to processes in engineering called *engineering processes*. Engineering processes are a subset of the business processes. Examples of engineering processes are the engineering change process or the process of product release. Similar to business processes in general, there is a real need and demand on reducing the time and costs of executing engineering processes in enterprises. In case of collaborative engineering or the development of a mechatronic product, there are additional needs for the integration of suppliers or engineering discipline specific processes. One approach to achieve this goal is to use common activity-driven workflow management systems (WFMS). They provide a set of constructs enabling the design of business processes in context to their environments. The activities are put in the foreground and the connection between the manipulated product data and the processes must be defined manually [Müller 2003]. This approach leads to three layered IT-Architectures in enterprises. On the bottom reside the systems responsible for the persistence of the product data. On the second layer, product data are manipulated and accessed via middleware technologies. On the top reside the workflow systems responsible for the modeling and management of the processes. However, in product engineering, the product data are tightly correlated with their related processes. The product data can only be correctly interpreted if they are considered in association with their processes. Hence, the correlation of processes and product data must be considered when modeling engineering processes. The strong separation of the process from the product data is a contradiction to this perception. In particular, every change of the product data structure can lead to manual changes of the associated engineering processes.
The introduced concept of Engineering Process follows another paradigm. The object orientation defines a process as a dynamical behaviour of an object. Processes are mapped to object methods in order to be executed. Engineering Processes are modelled following this paradigm and reside on the same layer as the Engineering Objects (See Figure 4).

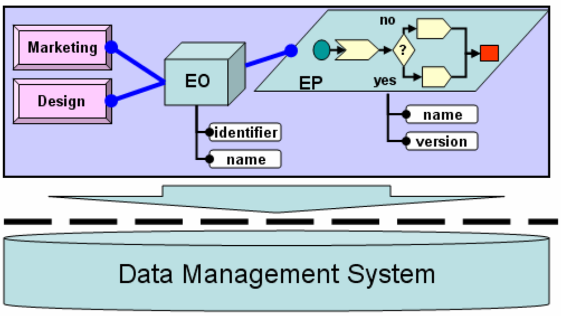

Figure 4. Example of an Engineering Network model

This approach breaks down the existing Three-Layer-Architecture of enterprise's information systems toward a Two-Layer-Architecture, where the product data and their processes are modelled and managed by the same data management system. So, the current available functionalities of conventional data management systems, like versioning and recovery, also apply for Engineering Processes.

3. Realisation and related work

In this section, approaches for the realisation of Engineering Networks are discussed. Realising EN requires the provision of modeling constructs enabling the definition of product data models and engineering process models following the concepts of Engineering Object and Engineering Process. The object-oriented paradigm and the standardized object-oriented modeling language UML (Unified Modeling Language) have become de facto standard for the design and implementation of Software system during the last decade [Engels 2000]. The UML is kept so general that it can be used to model any thoughts or concepts. But this is the source of its weakness, as concepts of specific domains cannot be expressed in fine detail. An answer of this weakness was the introduction of the concept of UML-Profile in UML. With this concept, the UML offers the possibility to extend its meta-model and thus, to support domain specific needs. The definition of a set of UML-profiles to realise Engineering Networks offers several advantages. First it can be straightforward implemented in current modeling tools supporting the definition of UML-profiles. Second it enables the application of the MDA approach to create executable code. In the sections below, the necessary UML-profiles for Engineering Networks are shortly discussed.

3.1 Realisation of Engineering Object

The concept of Engineering Object can be seen as an extension of the concept of objects in the object-oriented paradigm. The realisation requires a modeling support for its additional features virtuality, federation, graphical as well as services interfaces. The virtuality is based on the concepts of views and viewpoints. The concept of views is already used in product data management and in the database technology as an important mechanism for providing logical data independence, data hiding and is also a mean for data integration. However, many approaches to realise views consider them as a filter on a global model. This makes the management and the definition of relationships between views very difficult. We want to follow another approach considering a view as a part of an Engineering Object. That means, a view should now be handled as a normal attribute of an EO. Similar approaches to introduce view in the object orientation are followed in the research. [Nassar 2005] and [Hair 2004] have proposed a viewpoint oriented extension of the UML called View based Unified Modeling Language (VUML). This UML-profile provides a formalism for modeling software systems through objects and viewpoints. A generator for the automatically generation of Java code from the VUML following the MDA paradigm has also been implemented and tested. The concepts used in VUML offer a good basis for the realisation of the virtuality of Engineering Object. The realisation of the federated structure of EOs should also been supported by an adequate UML-profile. This profile should provide among other things constructs for a) making abstraction of application systems which contain the distributed properties using the concept of abstract data source, b) supporting the distribution of these properties over the defined abstract data sources, and c) supporting the definition of mapping between properties from different abstract data sources. The term abstract data source is a construct used to represent an application system as storage for distributed Engineering Object properties. It makes an abstraction of the location, the underlying hardware and software platform and the middleware technology used to access this application.

3.2 Realisation of Engineering Process

The modeling and management of engineering processes in enterprises is not a new topic. Many approaches have already be discussed and adopted in research and practice. The use of an *activity-driven* approach has already been considered as inadequate (see section 2.2). According to current research done at the VPE, the *data-driven* approach seems to offer a good basis for the realisation of Engineering Process. In the data-driven approach, the execution of a process depends on data feed into the process. Therefore it is one step to the integration of EOs and EPs to ENs.

4. Example of application field for Engineering Networks

An example for the application of Engineering Networks is the field of requirements engineering and management (RE&M).

The requirements structure represents a possible view on a product. The requirements view itself represents different subviews of different stakeholders (e.g. the customer needs, laws and standards, etc.). The requirements of a product are commonly presented in a hierarchical structure which follows a certain classification (e.g. classification by importance, classification by stakeholder, etc.). Hence requirements usually are not fixed or known at the beginning of a product development process, they are subject to a more or less complex change and refining process which takes place parallel to the product development process (see figure 5).

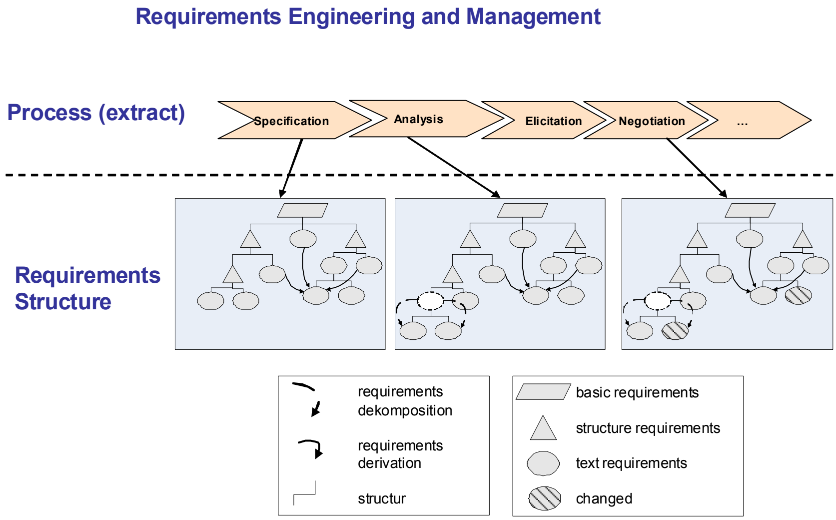

Figure 5. Requirements engineering process [KARE 2001] and the requirements structure

During the product development process, requirements are refined, decomposed, or new requirements are derived from existing ones. This leads to a requirement structure where requirements, additional the hierarchical structure are linked to each other. In current RE&M-tools and –methods, the requirements engineering processes are defined separate from the requirement information set itself. But, in order to interpret the information stored in a requirement, requirement process knowledge is necessary. E.g. requirement being the output of a simple requirement decomposition (splitting up an existing requirement) of an accepted requirement is much more likely to pass the negotiation phase without changes, than a requirement which is derived from another accepted requirements by applying product knowledge. Furthermore the requirements engineering process, shown in figure 5, vary depending on the nature of the requirement they are applied to.
Using EN, the requirements, their properties, their associated processes, as well as their structure are bundled in a PDM-system and can be managed and interpreted in context to each other.

5. Conclusion and further work

In this paper, a new concept for modeling product data and the associated engineering processes models has been introduced. The use of Engineering Object allows to model federated and integrated product data models and therefore addresses current needs for product data management in enterprises. Additionally, the concept of views and viewpoints supports the definition and management of user access rights to product information. Furthermore the concept of Engineering Process, offers new perspectives and visions for modeling and management of engineering processes in enterprises. In Further work, the realisation of the proposed UML-profiles will be addressed. The definition of graphical and service interfaces of Engineering Object will also be considered. We have focused in this paper only on the modeling aspect of Engineering Networks. Further work will also address the implementation of Engineering Networks and the generation of executable code from Engineering Networks models using the MDA approach. We believe that an extension of the common object-oriented programming languages is needed to support these tasks.

References

Bitzer, M.; Eigner, M.; Vielhaber, M.: "IMPACTS OF DESIGN PROCESS CHARACTERISTICS ON THE SELECTION OF PLM ARCHITECTURES". Paper accepted for publication for DESIGN 2008: 10th International Design Conference, Dubrovnik, Croatia, 19-22 May 2008.

Bitzer, M., Eigner, M., Langlotz, M., " MANAGEMENT DECISION SUPPORT BY PLM SOLUTIONS-EXECUTIVE SUMMARY", Proceedings of 16th International Conference on Engineering Design ICED 07, 2007, pp. 443-444.

Engels, G., Groenewegen, L., " Object-Oriented Modeling - A Roadmap", Special Track at 22nd International Conference on Software Engineering, ACM Press, 2000, pp. 103-116.

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, 2007, pp. 689-690.

Hair, A., " U_VBOOM : UNIFIED ANALYSIS AND DESIGN PROCESS BASED ON THE VIEWPOINT CONCEPT ", Proceedings of the 6th International Conference on Enterprise Information Systems, 2004, pp. 217-224.

KARE, "Knowledge Acquisition and Sharing for Requirements Engineering", www.kare.org, last visited 2001.

Müller, D., Reichert, M.,Herbst, J., "Flexibility of Data-Driven Process Structures", Workshop on Dynamic Process Management (DPM 2006), Reicher, M., Verma, K., Wombacher, A.(ed.), Springer-Verlag Berlin Heidelberg, 2006, pp. 181-192.

Nassar, M., "Analyse/conception par points de vue : le profil VUML", Laboratoire Groupe de Recherche en Informatique et Mathématiques du Mirail (GRIMM - UTM), 2005.

WFMC, "Workflow Management Coalition - Terminology & Glossary", Document Number WFMC-TC-1011, Document Status - Issue 3.0, 1999.

Dipl. –Inf. Fabrice Mogo Nem
Research associate and PhD candidate
Department of Mechanical and Process Engineering/University of Kaiserslautern, Institute for Virtual Product Engineering, Prof. Dr.-Ing. Martin Eigner
P.O. Box 3049, 67653 Kaiserslautern, Kaiserslautern, Germany
Tel.:+49 631 205 2312
Fax.:+49 631 205 3872
Email: mogo_nem@mv.uni-kl.de
URL: http://vpe.mv.uni-kl.de