www.cadanda.com Computer-Aided Design and Applications

On the Development of New Modeling Concepts for Product Lifecycle Management in Engineering Enterprises

Martin Eigner[1] and Fabrice Mogo Nem[2]

[1]University of Kaiserslautern, eigner@mv.uni-kl.de
[2]University of Kaiserslautern, mogo_nem@mv.uni-kl.de

ABSTRACT

Fast changes in today globalized business contexts, stricter becoming product and environmental legislations, and increasing complexity of products and related nowadays flexible engineering processes are some of the drivers for a further consideration of today applied concepts of multidisciplinary Product Lifecycle Management (PLM) which leverages engineering discipline specific Product Data Management (PDM) systems as backbone for an integrated product and process management throughout the entire product lifecycle. Adequate concepts should be provided to support the integration, structuring, interrelation and synchronization of today's multidisciplinary product data, to support the efficient management of the high number of product and associated engineering process variants, and to enforce and consolidate today's applied concepts of change and configuration management across partner and engineering discipline borders. Moreover, qualitative PLM solutions should be made more affordable for Small- and Medium-Sized Enterprises (SME) which have become more and more involved in multidisciplinary product development. This paper outlines concepts for a holistic approach denoted *Engineering Networks* aiming at coping with the aforementioned challenges. These concepts are the blueprint for the current implementation of *Engineering Networks* in a research project.

Keywords: PLM, process management, Lean PLM customization.
DOI: 10.3722/cadaps.2010.203-212

1 INTRODUCTION

Engineering enterprises today have to cope with various challenges. Amongst others, the pressure of today competitive and globalized business contexts, the increasing complexity of products driven by the increasing number of product variants, which in turn is caused by customer needs for better and tailored products, and the introduction of Mechatronics force enterprises to reduce their product development time and cost by simultaneously accounting for always stricter becoming product and environmental legislations, and by guaranteeing their customers permanent product innovation, high product quality and tailored product choices. These challenges have led to the adoption today of many engineering IT tools aiming at supporting the product development process. In order to be more competitive and efficient, many enterprises collaborate together in a supply chain manner to share their skills, services and competencies. This results into the creation of collaborative and multidisciplinary virtual engineering networks between enterprises. Therein, many enterprises contribute to the overall development of a product each of them focusing on its main competencies. Nowadays, more and more Small- and Medium-Sized Enterprises (SME) are involved as first-, second- or x-tier suppliers in such complex multidisciplinary engineering contexts and are therefore also dared to overcome the associated challenges which were earlier mostly considered being related to only large enterprises having sufficient human and financial resources to overcome them.

Many efforts have been spent so far to cope with the challenges related with multidisciplinary product development. In case of mechatronical products development, a survey made by the AberdeenGroup Inc. has shown that 68% of mechatronical product manufacturers face the problem of the synchronization of mechanical and electrical design representations, 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 [4]. Further, another survey showed that, although the most significant under the top six challenges during mechatronical product development are: the difficulty to find and hire experienced system engineers respectively the lack of cross-functional knowledge (50%), and the early identification of system level problems (45%), 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 [3]. 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 [23]), mature, tailored and user-accepted supporting engineering and management tools have to be provided to ensure the successful development of qualitative and reliable products. From a PLM perspective, the most cumbersome gaps to bridge in this regard are the integration, federation, structuring, synchronization and management of disparate and complex engineering partner or discipline specific product data and related engineering processes, as well as the management of their configurations and high number of variants.

This paper outlines concepts contributing to the solving of the aforementioned challenges multidisciplinary enterprises currently face regardless their sizes. The approach proposed is holistic and generative since software generators for the proposed modeling concepts are also considered as contribution to the reduction of the high costs for implementing PLM solutions which still constitute a barrier to their dissemination especially in SME [5]. These concepts are currently still under development in a research project. The remainder of this paper is as follows: section 2 summarizes the requirements for a new approach. The state of the art and related works addressing similar issues are discussed in the subsequent section 3. The concepts making up our contribution are presented in section 4. In section 5, a conclusion is given and further works are addressed.

2 REQUIREMENTS FOR A NEW APPROACH

The main idea behind the approach proposed in this paper is the holistic consideration of the three related dimensions *Technology*, *Process* and *Human* in a PLM solution for multidisciplinary product development as depicted in Fig. 1 below. In such a PLM solution, humans are responsible for creating and manipulating product information. They need to be supported during the execution of their tasks by accessing in real time the right among and quality of data they need. The processes represent the enterprises business processes in general and their computerized representation as workflow. We restrict however our scope to only the processes related to engineering like the engineering release or change process. The last dimension represents the enabling technologies and IT systems supporting value creation and dissemination inside or across enterprises. These IT systems should be based on best practices and standardized technologies in order to facilitate their dissemination especially in SME. A centralized repository should be set up at the heart of the PLM solution into which relevant collaborative product information should be mapped in order to support product data synchronization, change and configuration across partner or engineering discipline borders. This repository could be physically part of a partner PLM solution and should not be necessary deployed as a physically stand alone system.

The main requirements towards a new solution derived from the challenges mentioned above and based on this consideration of a multidisciplinary PLM solution can be summarized as follows:

- **<u>Req 1:</u>** Provision of adequate modeling concepts and a modeling methodology enabling the integration and structuring of multidisciplinary product data following discipline, partner or even process step specific needs for information.
- **<u>Req 2:</u>** Provision of means for interrelating and synchronizing multidisciplinary product data in order to map and manage their semantic dependencies.
- **<u>Req 3:</u>** Integration of today's flexible engineering processes with manipulated product data in order to improve the management of their changes and high number of variants.
- **<u>Req 4:</u>** Application of configuration management across partner and engineering discipline borders and support for analyzing the impacts of engineering changes.
- **<u>Req 5:</u>** Support for a better customization and acceptance of PLM solution especially in SME.

Especially the three first requirements listed above define the scope of this paper.

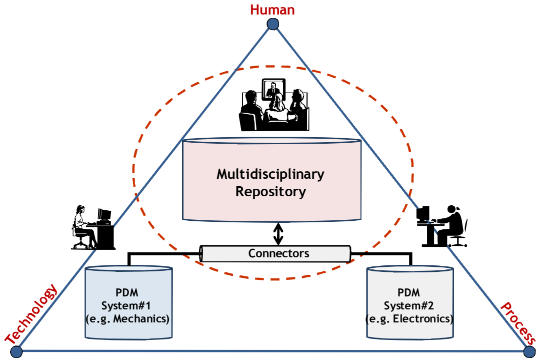

Fig. 1: Understanding of a PLM solution for multidisciplinary product development.

3 STATE OF THE ART AND RELATED WORKS

This section discusses the state of the art and points out works aiming at addressing similar issues. These are: multidisciplinary product modeling, integration, management and configuration of the high number variants of product and related engineering process data, and the support for better customization of PLM solutions especially for SMEs.

Enterprises use nowadays tailored and different PDM systems in the disciplines Mechanics, Electronics/Electrics and Software Configuration Management (SCM) systems in case of Software Engineering, for the management of product related information during product development. They also used 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 respectively used. In order to span the whole product lifecycle, the new paradigm of Product Lifecycle Management (PLM) has been introduced [9]. Several mature and even standardized product data and process models are nowadays available. One of the famous standardized object oriented product data models are 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. Instead, many research activities have proposed to integrate the standardized discipline specific product data models defined in STEP to a global integrated product data model by means of object-oriented aggregation [15, 24, 26]. Doing so, the discipline specific partial product data models are aggregated to the mechatronical product data model using view elements. This approach offers a good way to bring together discipline specific product data to form a mechatronical product. However, the usage of aggregation to link together the disparate partial product models has some disadvantages. First, the semantic associated with each discipline specific view is missing, making it difficult to semantically relate information belonging to separate views. Further, adequate algorithms need to be implemented to extract from the resulting information cloud the relevant data a user needs. We propose to use a more adequate object oriented modeling approach supporting the structuring of product data by considering the viewpoints of the users accessing it. Such a view-centered modeling paradigm has been proposed in [19, 11] for modeling complex software systems. We adapt this approach to support multidisciplinary product modeling for example by allowing the assignment of multiple views of an object to a user, by suppressing the definition of a *DefaultView* from which extended view definitions are derived, and by allowing a view to reference a subset of the object's properties which contains it.

Regarding the synchronization of multidisciplinary product data, studies show that semantic dependencies between data objects can be mapped to dependencies between their dynamical evolvement models [18]. The concept of lifecycle modeling as means to represent and manage the dynamical evolvement of an object or even an object relationship along its entire lifetime has been successfully applied to product data [7, 13, 18, 22]. An object lifecycle defines a set of states an object can go through during its entire lifetime. In the context of PLM, different access rights, versioning concepts or in general different behaviors can be defined for product data depending on its current state in its lifecycle [1]. The modeling and management of lifecycle models for product data ensure the correct deployment of processes operating on them and therefore, is a means to synchronize and manage dependencies between product data. To ensure compliance of object lifecycle and processes, [13, 22] propose an approach for generating a compliant business process model from a set of given reference object lifecycle models. This approach considers the unification of object lifecycle models by considering joint prescribed behaviors and ensures that invalid composite states can not be reached. Applied to mechatronical product data, this concept can be used for the definition of a unified lifecycle model for a mechatronical product based on the composition of the heterogeneous lifecycle models of its components. Another approach for modeling product dependencies using lifecycle models has been proposed in [18] as part of the COREPRO (Configuration Based Release Management Processes in the Automotive Sector) approach. The COREPRO modeling framework allows creating data-driven process structure as discussed in the paragraph to follow. It is based primary on the definition of lifecycle models and lifecycle dependencies, as well as the consideration of object structure to generate data-driven process structure. The two aforementioned approaches make use of object lifecycle models and dependencies to synchronize heterogeneous objects and ensure compliance with process structures. These approaches were not intended to support directly the synchronization of multidisciplinary product data. However, they provide fundamentals which can be used and extend in this regard. We propose to leverage the advantages of both approaches and provide constructs for explicitly modeling dependencies between lifecycles using Boolean formulas. This avoids the cumbersome manually modeling of composite lifecycle states as prescribed in [13, 22]. Further, we introduce a new kind of relationship between objects called LCD (Lifecycle Dependency) which expresses a dependency between the lifecycle of related objects. Doing so, we extend the approach in [18] to express dependencies between objects which are not necessary directly related by means of object oriented aggregation.

The modeling and deployment of business processes have been used for long time now to solve the weakness, inflexibility and traceability lacks observed during the era where these processes were hidden in application logics. In the last years, paradigm shifts have been observed regarding the way processes are modeled. In opposite to the traditional activity-centric approach of process modeling where the tasks and activities to be executed are put in the foreground, new paradigms have emerged putting both activities and manipulated object structure in the center of interest during process modeling [6, 18, 20]. Examples are the PBWD (Product Based Workflow Design) and data-driven process modeling approaches. The former allows the specification or redesign of optimized process models based on the bill of materials of a product [20]. The later addresses the modeling and coordination of complex processes operating on complex products, which are made up of several sub-processes, and where the interrelation of these sub-processes relates to the assembly structure of the complex product. Doing so, the dependencies between the lifecycle model of the product and those of its components are used to generate according process structures [18]. PBWD and activity-driven process modeling share in common the exploitation of similarities between product and process structures. However, both do not address the way such a tight relation between a product and its engineering processes can be exploited to jointly manage them and therefore to improve traceability of their changes. Further, engineering processes are often modeled based on widespread best practices and due to the enterprise specific adaptations of these processes and their evolutions, it has become indispensable to improve the management of resulting high number of process variants which are still separately modeled and manually maintained [2, 16, 25]. An operational approach for managing process variants based on a single process model has been proposed in [12]. Some process modeling languages like C-EPC (Configurable Event-Driven Process Chain) provided means for representing configurable process models which incorporate several process variants into a single process model using variation points [21]. Thereupon, a questionnaire-based variability framework to link domain configuration to process configuration has been proposed in [14] in order to automate the configuration of a single process variant by answering to a set of questions which together define the domain scope. This approach is however based on a configurable business process modeling language and the domain scope is not necessary related to product development.

Introducing and customizing today's PLM solutions in enterprises remain a cumbersome and cost expensive issue. The costs for introducing a PLM solution are commonly made up of expense for one-time consulting services generally needed for analyzing the enterprises processes, identifying the requirements and preparing the enterprise for the PLM solution introduction, the expense for required software licensing including the PDM system at the heart of the PLM solution, the expense for required hardware, the expense for implementing the enterprise specific requirements to the PLM solution identified during requirement analysis denoted as *Customizing*, and miscellaneous expenses for training and documentation. Although efforts have been spent by providers of PLM solution in order to address the problems of SME by providing focused solutions based on pre-packaged functionalities, these solutions should also be made easily tailored (customized) to fit the specific needs of an enterprise [5]. Empirical observations show that alone the costs for customizing a PLM solution during it introduction in a large enterprise can reach the fourfold of the intrinsic costs for required software licensing. This 1/4 ratio of licensing to customizing cost observed during the introduction of a PLM solution is obviously worse in SME due to the relatively lower number of required software licensing. More attention should therefore be paid to address the lean customization of PLM solution especially during their introduction in order to facilitate their implementation in SME and get the solution into production as quickly and cheap as possible.

4 THE PROPOSED SOLUTION IN ENGINEERING NETWORKS

This section outlines how especially the three first requirements depicted in section 2 have been addressed in *Engineering Networks* [8, 17] which is currently still under development.

4.1 Modeling Multidisciplinary Product Data Using Engineering Objects

The modeling concepts described here are intended to address the first requirement (**Req 1**). Their fundaments are the extension of current object oriented product data modeling approaches with the concepts of *Property*, *View*, *Viewpoint* and *Lifecycle Dependencies (LCD)* as part of a new type of objects denoted *Engineering Objects (EO)*. Considering the two modeling abstraction levels *Model* and *Instance* according to the Meta Object Facility (MOF) defined by the OMG (Object Management Group), an EO represents at *Instance* level an (multidisciplinary) object within a given state in its lifecycle (e.g. a whole car with product number 4711 and which has been released for manufacturing) and it is defined at *Model* level by an EO-Type (similar to a Class in term of object orientation).

The concept of *Property* of an EO is used to represent properties of products during the Product Generation Process (PGP) [10]. During the PGP, end-user relevant product properties must have the highest priority and in parallel, engineering, economical as well as environmental properties significant for the product also have to be captured and must be appraisable to the control of the PGP. These product properties are at present only captured in listings while a formal structure with the dependencies and relations between them is still missing. Product properties can be extracted during the requirement analysis phase and extended during subsequent phases in product development. Some of them formalize the targeted product qualities (e.g. the maximal velocity, the weight or the CO_2 emission of a car) and influence the engineers' thinking. Especially in context of multidisciplinary product development, some properties can be seen as engineering discipline neutral and therefore as central collaboration items. We distinguished during the formalization of EO's properties in order to allow their management inside an IT-System between nowadays already considered static product properties such as *name*, *lifecycle state* or *version number* which preliminary serve administrative purposes, and those EO's properties of particular interest for the PGP for which dependencies with others can be defined and controlled. We therefore used different meta-classes for realizing them.

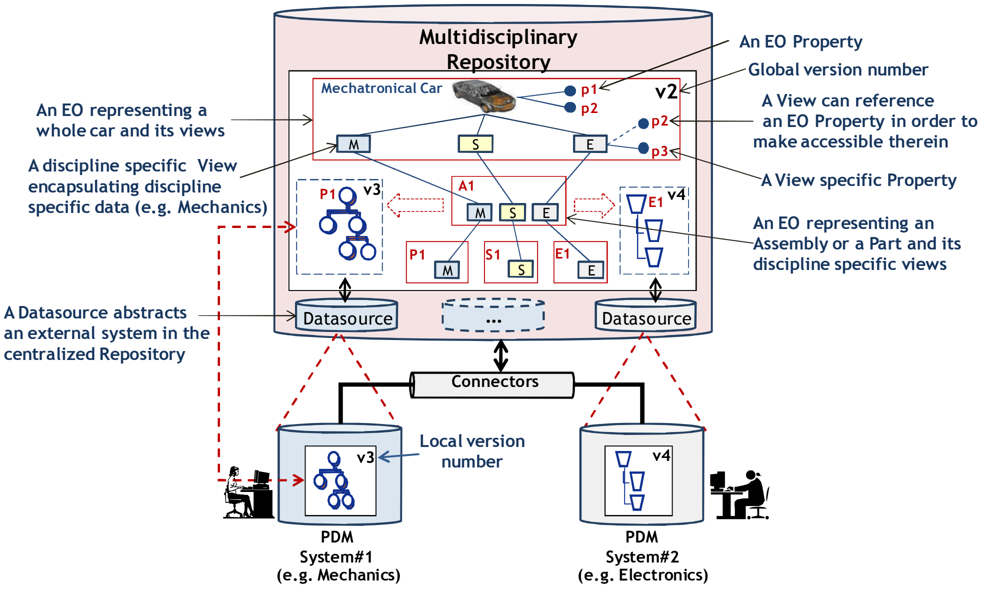

Fig. 2: Deeper view inside the centralized repository at the heart of a collaborative PLM solution.

The concept of *View* is used for integrating, structuring and abstracting the scattered information making up a whole EO. A view encapsulates information relevant e.g. for a specific partner, a specific engineering discipline or even a process step manipulating this EO. Considering the example in Fig. 2, which gives a deeper insight into an EO-based realization of a centralized repository as depicted in Fig. 1, an EO is used at top level to represent a mechatronical car which is being developed in a multidisciplinary engineering context. On this car, various views can be defined in order to structure at higher level its discipline respectively partner specific product information. Represented views are: a Mechanical (M), an Electronics/Electrics (E) and a Software (S) view. The discipline specific product information is extracted from local IT- Systems and mapped into the repository. In order to illustrate how information are mapped into the repository using several EOs, let's consider the fragment of the mechatronical Bill of Material (BOM) for the car which corresponds to the structure of the built-in wiper (A1) and its discipline specific constituents: the mechanical box (P1), the printed circuit board (E1) and the controlling software (S1). These four EOs have appropriate view definitions encapsulating discipline specific information and related structure. The information that the wiper (A1) consists of the mechanical Part P1 is represented using a relationship between its mechanical view M and the corresponding view in P1. Doing so, once a person having restricted access privileges to only the mechanical information pertaining to the car accesses it, only the information in the views M and the related EOs such as A1 and P1 are displayed. 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, this perception assimilates a view to a kind of filter on a global model making the definition of dependencies between views difficult. The Standard STEP also provides means for defining views on a product version. This approach uses aggregation to link views to product versions. In Engineering Networks, a view is considered as integral part of an EO, and it is defined at modeling time using EO-Type. Once an EO of a specific EO-Type with view definitions is instantiated, memory is allocated dynamically for each activated view of this EO. An EO can therefore be thought of as a kind of high intelligent object having enough information about itself and able to behave differently depending of the current user manipulating it. A view can also references properties of the containing EO or can define its own. Consequently, the total set of properties for an EO is the union of its view specific properties and those directly associated to it. The semantic associated with views are very useful to enable the definition of semantic relations between EOs with similar view definitions.

Another modeling concept used by the definition of EO-Types is the *Viewpoint*. A viewpoint is the combination of multiple view definitions of an EO-Type and can be assigned to a user or a group of users through roles and role groups. It represents the sight of a user on instanced of this EO-Type. This sight can be for example discipline, partner or application specific. This concept on the top of the concept of view is very useful to support the definition of user specific needs for product data. We make additional use of these concepts of *View* and *Viewpoint* to support the user specific generation of adequate Graphical User Interfaces (GUI) to manipulate EO's information (see Section 4.5).

4.2 Modeling Multidisciplinary Product Data Dependencies

This section introduces the concept of *Lifecycle Dependencies (LCD)* used to map and manage part of the dependencies between multidisciplinary product components and synchronize their collaborative development. Through this concept of LCD the second requirement (**Req 2)** is addressed.

Several models for modeling object lifecycle are available today. They share in common the definition of states, transitions between states with associated transitions constraints, and pre-/post-action for transitions. Transition constraints are typically expressed using user privileges (e.g. only user with specific privileges can promote an object to its next state) or using events (e.g. an object of a specific type is automatically promoted to a specific state upon deletion). In current PDM systems, it is usual to define specific access permission or versioning model for an object or an object relationship in a given state (e.g. upon a document has reached its release state, it is only accessible by a specific group of users and can no more be changed) [1]. We extend this common perception of object lifecycle with the expression of lifecycle dependencies. A lifecycle dependency expression (LCDE) can be used as guard or trigger for a transition. When used as a guard, a LCDE prevents a transition to occur before the condition is fulfilled. When used as event, a LCDE can trigger a lifecycle change upon it has been evaluated to true by the managing IT-System. Before such a LCDE during the specification of the lifecycle model of an EO-Type can be expressed, a directed LCD relationship must be defined previously from this EO-Type to the EO-Types whose lifecycle states are used for expressing the dependency. This LCD relationship denotes the existence of semantic dependencies between the source and target EO-Types. In this case, the lifecycle of the source EO-Type can be influenced by the lifecycle of the related EO-Type. For example (see Fig. 3.), if a mechatronical component can only change its state from an initial state S1 to a new state S2 in its lifecycle when associated mechanical and electrical components have already reached respectively their states 'MP' and 'EP', a LCD can be defined between the EO-Types representing the mechatronical component and those representing respectively the mechanical and electrical components. The semantic of the LCD is expressed using a Boolean expression. This transition can only occur if this guard evaluates to true.

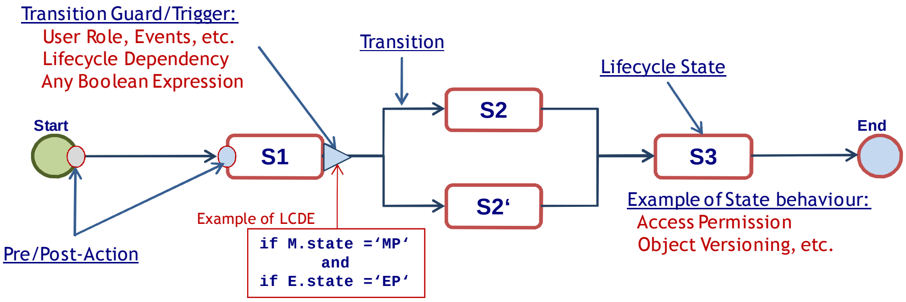

Fig. 3: Modeling lifecycle dependencies as mean for product data synchronization.

4.3 Integrating and Configuring Product and Related Engineering Process Data

The support for the enactment of core engineering processes such as the process of engineering change by PDM Systems is considered today as being a fundamental requirement. However, although configuration management has been successfully applied on discipline specific product data in order to allow the retracement of their historical evolvement over time in case of product reconfiguration, little attention has been paid so far on managing and configuring in PDM Systems the today's flexible and high-variant engineering processes manipulating the product data. The approach outlined here is intended to holistically address this shortcoming and therewith the requirement **Req 3**.

The global overview of the approach proposed in Engineering Networks is depicted in the Fig.4. At the *Semantic Level* we use the previously introduced concept of *Engineering Object* for modeling, structuring and integrating heterogeneous product data scattered over various IT-Systems at *System Level*. At the *Instance Level* the concrete product instances as sold to customers are manipulated (e. g. a real instance of a car which is sold to a specific customer). Between the *Instance Level* and the *Semantic Level* resides the *Variant Level*. The later is nowadays successfully used for efficiently managing the high amount of product variants using a product variant structure. Product configuration is used for bridging the gap between *Variant* and *Instance Level* (a customer configures during ordering its individual product variant by selecting appropriate items from a list of predefined product options also called product specifications). The reverse step is the product reconfiguration.

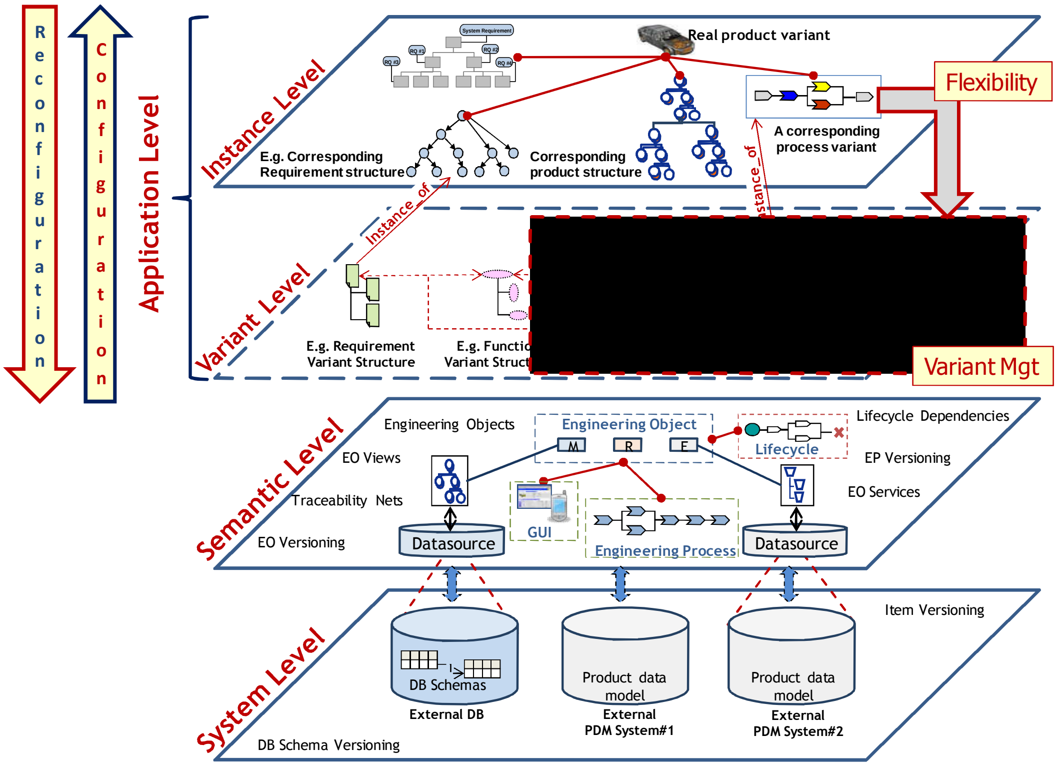

Fig. 4: The holistic modeling approach proposed in engineering networks.

In Engineering Networks we would like to go further by applying these well established concepts on engineering processes in order to increase the traceability of both product and process information. This is achieved by tightly integrating at both *Semantic* and *Variant Level* product and engineering process data. Doing so, we choose to follow up the approach for process configuration proposed in [14]: by using the product variant structure as a domain model for process configuration, by defining a variant structure for engineering processes and therefore extending the scope our approach to those traditional process modeling language without direct configuration support, and by using product variant selection code to configure the associated engineering process variants. In order to keep such a variant structure for engineering processes up to date, changes on engineering process models (regardless its nature: e.g. evolutionary changes or ad hoc changes during enactment in case of a flexible process deployment) should be reflected in the process variant structure at *Variant Level*.

Doing so, product and related engineering processes are tightly integrated and commonly managed in the same repository [2]. Hence, given a product configuration, it should be possible not only to trace its development history but also the history and variants of the engineering processes it walked through reaching this configuration. Furthermore, currently applied concepts of configuration management based on a thoroughly implemented change management are applied in the centralized repository on both product and process data at both *Variant* and *Semantic Level*. In order to ease the analysis of the impacts a change on a product component will have across partner or engineering discipline borders, the available information about Lifecycle Dependencies stored inside the centralized repository are intended to be used in order to build traceability nets between multidisciplinary data and therefore as means for addressing the requirement **Req 4**.

4.4 Addressing the Lean Customization of PLM Solution for SME

Implementation and customization of PLM solution constitute also a barrier to its dissemination in SMEs. For supporting the development of easy customizable PLM solution, we proposed first to enrich the underlying modeling concepts for multidisciplinary artifacts using the new object oriented modeling concept denoted *Engineering Object*. Providing more powerful and adequate modeling expressiveness contributes to the reduction of customization efforts. Further, we propose to build on existing standardized and open source solutions to implement a framework consisting of five modeling tools and several generators which are responsible for creating, editing and deploying the models addressed in *Engineering Networks* [8]. Doing so, the requirement **Req 5** is addressed.

5 CONCLUSION

This paper has outlined a new holistic approach for multidisciplinary product modeling. The goals of this approach are the support for integrating, structuring and synchronizing multidisciplinary product data, the support for integrating product data with related flexible engineering processes, the support for managing the high number of variants of nowadays flexible engineering processes, and the support for a better customization of PLM solutions. The proposed modeling concepts have been the blueprint for the specification of a framework consisting of five modeling tools and several generators [8]. Future works will address the implementation of this framework and a required modeling methodology accompanying it. Both are currently subject of a project. The implementation of the framework as proof of concept could eventually induces some minor changes to concepts presented in this paper. Advances in the implementation or any changes to the concepts will be also addressed in future papers.

ACKNOWLEDMENTS

This paper is based on work achieves 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

[1] Open Source PLM Solution - Aras Innovator, http://www.aras.com/.
[2] Bea, H.; Bae, J.: A Version Management of business Process Models in BPMS, Lecture Notes in Computer Science (LNCS), 4537, 2007, 534-539.
[3] Boucher, M.; Houlihan, D.: System Design: New Product Development for Mechatronics, Aberdeen Group, 2008, http://www.aberdeen.com/.
[4] Chad, J.: The Mechatronics System Design Benchmark Report - Coordinating Engineering Disciplines, AberdeenGroup, 2006, http://www.aberdeen.com/.
[5] CIMdata: Manufacturing Enterprises of All Sizes can benefit from PLM-PLM has Arrived for Mid-Size Manufacturing Enterprises, Whitepaper, 2003, http://www.cimdata.com/.
[6] Cohn, D; Hull, R.: Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges, Proceedings of OTM 2008 Confederated International Conferences, Monterrey/ Mexico, 2008, 1152-1163.

[7] Ebert, J.; Engels, G.: Specialization of Object Life Cycle Definitions, Fachberichte Informatik 19/95, University of Koblenz-Landau, 1997.

[8] Eigner, M.; Mogo Nem, F.: Conceptual modeling and generator framework for multidisciplinary and collaborative Product Lifecycle Management, Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design, Santiago/Chile, 2009, pp. 590-595, ISBN: 978-1-4244-3535-7.

[9] Eigner, M.; Stelzer, R.: Product Lifecycle Management; Ein Leitfaden für Product Development und Lifecycle Management, Second Edition, Springer Verlag, 2009, ISBN: 978-3540443735.

[10] Faisst, K.-G.; Dankwort, C.: New Extended Concept for the Usage of Engineering Objects and Product Properties in the Virtual Product Generation Process, Proceedings of the 16th International Conference on Engineering Design: ICED 07 - Knowledge, Innovation and Sustainability, Paris/France, 2007, 689-690, ISBN: 1-904670-02-4.

[11] 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, 217-224.

[12] Hallerbach, A.; Bauer, T.; Reichert, M.: Managing Process Variants in the Process Lifecycle, Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS'08), Barcelona/Spain, 2008.

[13] Küster, J. M.; Ryndina, K.; Gall, H.: Generation of Business Process Models for Object Life Cycle Compliance, in: Business Process Management, Volume 4714/2007, Springer Berlin/Heidelberg, 2007, 165-181.

[14] La Rosa, M.; Gottschalk, F.; Dumas, M.; Van der Aalst, W. M. P.: Linking Domain Models and Process Models for Reference Model Configuration, Proceedings of the 10th International Workshop on Reference Modeling LNCS 4928, Brisbane/Australia, 2007, 417-430.

[15] MechaSTEP – STEP data model for simulation data of mechatronic systems, DIN PAS 1013, Beuth-Verlag, 2001.

[16] Mogo Nem, F.; Eigner, M.: Common versioning of product data and engineering processes, in: Chakrabarti, A. (Hrsg.): Research into Design - Supporting multiple Facets of Product Development, Bangalore/India, 2009, 295-302, ISBN: 978-981-08-2277-4.

[17] 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/Kroatia, 2008, 849-856, ISBN: 953-6313-91X.

[18] Müller, D.; Reichert, M.; Herbst, J.: A New Paradigm for the Enactment and Dynamic Adaptation of Data-driven Process Structures, Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE'08), Montpellier/France, 2008, 48-63.

[19] 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, 257-265.

[20] Reijers, A. H.; Limam, S.; Van Der Aalst, W. M. P.: Product-Based Workflow Design, Journal of Management Information Systems, 20(1), 2003, 229-262.

[21] Rosemann, M.; Van der Aalst W. M. P.: A Configurable Reference Modeling Language, Information Systems Volume 32, 2007, 1-23, ISSN: 0306-4379.

[22] Ryndina, K.; Küster, J. M.; Gall, H.: Consistency of Business Process Models and Object Life Cycles, in: Models in Software Engineering, 4364/2007, Springer Berlin/Heidelberg, 2007, 80-90.

[23] Verband Deutscher Ingenieure / Association of German Engineers: Design methodology for mechatronic systems, VDI 2206, 2004, http://www.vdi.de/.

[24] Waterbury, S. C.: STEP for Multi-Disciplinary Model Management: Intelligent PDM, NASA/Goddard Space Flight Center, 1999.

[25] Xiaohui, Z.; Chengfei, L.: Version Management in the Business Process Change Context, Proceedings of the 5th International Conference on Business Process Management, Lecture Notes in Computer Science (LNCS) 4714, 2007, 198-213.

[26] Zhu, S.; Cheng, D.; Xue, K.; Zhang, X.: A unified Bill of Material Based on STEP/XML, Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design, Nanjing/ China, 3, 2006, 267-276.