CHANGES ON CHANGES: TOWARDS AN AGENT-BASED APPROACH FOR MANAGING COMPLEXITY IN DECENTRALIZED PRODUCT DEVELOPMENT

Kehl, Stefan (1); Stiefel, Patrick (2); Müller, Jörg P. (1)
1: Clausthal University of Technology, Germany; 2: Volkswagen Group

Abstract

In this concept paper, we report on ongoing work aimed at a novel approach to developing complex products. Based on an analysis of the requirements of product development in the automotive industry, the main problems we observe are limited transparency, patency, and reuse. These problems are even more pronounced - and more difficult to manage - in multi-brand product development settings, with spatially distributed and organizationally autonomous development partners.

In order to decrease the amount of information each actor has to manage, we propose and illustrate the novel notion of virtual product model components (VPM-C) as an approach to address these challenges. We propose a conceptual architecture of virtual product models, which supports four concepts (views) as first-class citizens: parts, geometries, features, and processes. To handle the dependencies between elements of a VPM-C, we further suggest an agent-based approach, and outline a corresponding architectures and design alternatives. We illustrate these basic concepts by use case scenarios derived from an analysis of automotive product development practices.

Keywords: Collaborative design, Complexity, Design practice, Active components, Multi-Agent Systems

Contact:
Stefan Kehl
Clausthal University of Technology
Department of Informatics
Germany
stefan.kehl@tu-clausthal.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

At the beginning of the 1990s most automotive manufacturers offered about eight different products (VDA, 2012). In order to meet the increasing customer demand for customizability the number of derivatives of each product is steadily increasing. This trend is often referred to as *mass customization*. According to (Tseng and Jiao, 2007), mass customization is about “producing goods and services to meet individual customer’s needs with near mass production efficiency”. Some practitioners propose concepts to increase the commonality of multiple products and product lines regarding their individual components (Benavides *et al.*, 2010). In order to utilize those synergistic effects in different product families as well (Winterkorn, 2009), this leads to an increasing modularization of complex products (Clarkson *et al.*, 2004) (Jarratt *et al.*, 2011). Therefore the *overall product* (respectively the resulting product line) is far more complex than the individual products themselves, as the complexity of a product is related to the connections between its components (Simon, 1996) and each component is connected to others even in different products (of the product line).

Furthermore, products are rarely developed from scratch, but are based on previous models. Therefore the development of a new product is not a linear process, but a continuous sequence of changes on components of existing products (Bucciarelli, 1994), (McMahon, 1994), (Cross, 2000), (Cheng and Chu, 2012) over the entire lifecycle of a product (Jarratt *et al.*, 2011). Each change on a component may have an impact on other components which may be connected to several others and so on. In this context (Fricke *et al.*, 2000) talk of an *avalanche of change*. Obviously this effect multiplies according to the number of connections and therefore the size of the product line.

Considering the product development process as such a sequence of changes, we can state three crucial requirements:

- The support of **reusability**, meaning that a component can be assigned to multiply products. Each of these assignments is defined by context specific information regarding the usage of the component in that specific context (e.g. the position of the four rims depends on the length of car).
- The **patency** of a product model describes the uninterrupted flow of information from early stages of the product development process to later ones and from one domain to another. According to (Clark and Fujimoto, 1991) (Jarratt *et al.*, 2011), “up to two-third of technical changes could be prevented by better communication”.
- **Transparency** referring to the preservation of consistent states regarding the connections between components, components themselves, and the traceability of changes over the entire product development process.

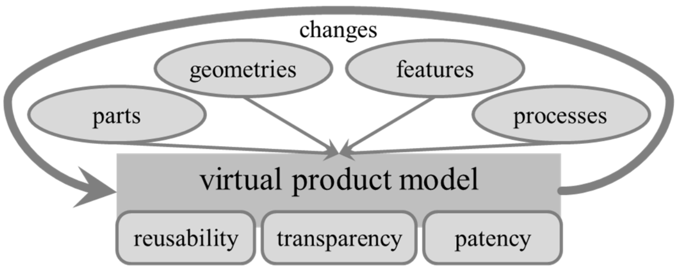

Figure 1. A holistic view on a virtual product model

Changes on a component as well as its initial design are performed by experts with different roles. In order to divide the amount of information each of these experts has to manage, we propose a **virtual product model (VPM)** where each component consists of the following four elements (see also Figure 1):

- **Parts** that carry organizational information such as suppliers and procurement channels.
- **Geometries** carry the information about the geometric characteristics of a **part**. They are usually represented by CAD-Files. Thus the part dependents directly on its realizing geometry.
- **Features** such as technical descriptions of individual components and
- **Processes** that use this component (e.g. production processes in different plants)

We are, to our knowledge, the first to conceptualize a product component (and the resulting virtual product model) in this way.

Much related work focuses on the engineering perspective of the product development process (Clarkson *et al.*, 2004) (Jarrat, T. A. W. *et al.*, 2004) (Jarratt *et al.*, 2011). In particular, much attention has been paid to the field of engineering change management or the impact of geometrical changes of components on each other and the overall product (Hamraz et al., 2012). In contrast, we investigate the coordination of changes on components made in different disciplines. E.g. due to supply difficulties the procurement department cannot procure a specific material *A*. Another material *B* fulfils similar characteristic but is not as robust as *A*. In order for the component still to comply with the requirements (regarding load-bearing capacity), the engineering department has to alter the geometrical representation.

By presenting our experiences during investigations on the current practice in the automotive industry, we aim to contribute to fill this gap. Furthermore, we illustrate our conceptual approach for an agent-based coordination of changes on an integrated virtual product model.

The structure of this paper is as follows: In Section 2, we describe some real-world use cases from the automotive industry. In Section 3, we identify and discuss sources of complexity in the management of cross-brand product development projects illustrated by the use cases in Section 2. Section 4 introduces our conceptual approach for dealing with cross-organisational change management taking into account the described use cases. Finally, we conclude with a summary and the description of our future work.

2 AUTOMOTIVE PRODUCT DEVELOPMENT USE CASES

In the following, we describe use case scenarios from the perspectives of different stakeholder within the product development process. Each subsection corresponds to one of the examined perspectives.

2.1 Project administrator

The person responsible for a development project wants to know whether the specifications from the product definition (early stages of the product development process) have been obeyed. As products are rarely developed from scratch, some components of the previous model are carried over to a new one. These components can be defined as *fixed* (no alteration during the following development process) or as a *basis for further changes*. E.g., the project administrator defines that the exterior mirrors of a *product i* are the basis for the exterior mirrors of another *product i+1* and provides a specification that describes the alterations that will be made to the geometric representation of these mirrors. A designer changes the geometry according to that specification.

Furthermore, the project administrator is interested in how many components have been reused in the new product. In other words, the degree of reuse over several products has to be determined.

In summary, from an administrative point of view, the VPM has to be transparent regarding the reuse of the individual component.

2.2 Purchaser

Considering the same rim being assembled at plant A and B, but delivered by different suppliers, this results in two parts that are geometrically identical, because parts contain the information about procurement channels. Therefore, a purchaser has the following queries on a VPM:

- Which parts can be procured together (because they are geometrically identical)?
- Are there alternatives for part X (because the supplier cannot deliver on time)?

2.3 Designer

Designers are mainly interested in answers to the questions whether there are existing geometries that fulfil their requirements and how they can pass on the results of their work without any extra effort.

- For instance, a designer receives the order to design a specific component (geometrical representation). The specification of can be used to retrieve suitable component of the virtual product model (VPM-C) hat are stored in a shared repository. The result is a VPM-C, which was developed as a part of a different product and meets most of the specifications (in the best case all of them). If the VPM-C does not meet all requirements, the designer alters the geometry according to the given specification.

After that, the designer needs to check in the new version geometry. Ideally, this is done by just pressing a button because all information for the following processes is already present in the system.

2.4 Process planner

In contrast to the roles mentioned above, the process planner is not involved in the development but in the manufacturing of the product. Therefore, the main questions a process planner is dealing with are not related to the definition of the component but to its further use:

- Are there manufacturing processes for component A in plant B?
- How does the assembly of similar parts work?

If a geometry is reused, the corresponding manufacturing or assembly processes are also known, because they belong to the same VPM-C. Therefore, the process planner could use this information directly or (at least) as a basis for further planning. Alternatively, if the geometry has no processes attached, the process designer needs to enquire how similar components are assembled.

2.5 Use cases across different views

Since diverse roles are involved in the development of a product (and each component), there are use case scenarios that describe the interactions of these roles.

For instance, let us consider the design of a rim: In this process, usually the geometry (3D representation) is the first element of the corresponding VPM-C that exists in the product model. Therefore, the designer, who is only interested in those geometric representations, submits his CAD-File (computer-aided design) of the rim. Because a rim is installed not just once, but four times in a vehicle, the product planner defines these usages (front left, front right, rear left, rear right) of the rim. This information may be defined in the BOM (bill of materials). Therefore a part (representation of a component in the BOM) and its usages need to be defined in some kind of part model. Furthermore there are usually several types of rims for a vehicle. Therefore variances regarding all other possible rims are defined on the part respectively its usages. During a later phase of the product development process (shortly before the start of the production) this rims need to be procured respectively produced taking into account the quantity (four usages of the same part).

3 SOURCES OF COMPLEXITY

Understanding each version of a specific component as a (consistent) state and the development of a new version as the transition from on state to another, the complexity of creating and managing virtual product models is mainly influenced by the following three factors:

- **Data Model:** In order to plan, validate, and build products they have to be represented. The data model stores all information concerning components of the product in all phases of the product development process.
- **Organization:** The allocation of competences and privileges within a development project.
- **Processes:** Each data model object exists for a purpose constituted by the processes used to manipulate the information stored by the data model objects.

In the remainder of this section, we will elaborate on the impact of these complexity sources on the VPM and draw conclusions resulting in requirements guiding our further research.

3.1 Data model and changes

Parts, geometries, processes, as well as *feature*s have their own life cycles that are controlled by different actors (designer, purchaser or logisticians e.g.) from diverse disciplines. Designers create new or modify existing *geometries* and thereby create new versions of the VPM-C continuously. Purchasers or logisticians react on changing market conditions by changing *part* information. The *manufacturing processes* in turn (design of the production facilities, processing order, etc.) are specific for concrete geometric characteristics of a component but also depend on local factors. Considering a product manufactured at two different plants, the individual manufacturing processes may be different due to the availability of resources or spatial limitations of the production site. Therefore there are most likely one-to-many relationships between geometries and multiple manufacturing processes. These processes are also subjects to ongoing adjustments due to changing location factors during their life cycles. This contributes significantly to the increase in complexity of the virtual product model.

The next information category contained in a VPM-C is called *features* such as the technical description of the component. E.g., if a new specification leads to a change on the component's material, this results in a change on the part-information. This in turn can lead to the need of adjusting the geometric characteristics, which may lead to altered manufacturing processes (see Section 2.2). A direct impact of the technical description on the geometry is conceivable, too. Such a trigger could for example be a specific customer requirement in terms of the product's look and feel.

In summary, this means that each component of the virtual product model consists of four information carriers with independent lifecycles. These carriers, however, are mutually dependent. If we consider a consistent state (regarding the relationships between the elements of a VPM-C) as a version of a VPM-C and the coordinated adjustment of at least one information carrier as a state transition, the sequence of state transitions can be considered as an individual life cycle on the VPM-C itself. This understanding of the life cycle of a VPM-C as *changes on changes* is illustrated in Figure 2.

We are, to our knowledge, the first considering a component (and the resulting virtual product model) in this way. Especially with regard to post change processes (e.g. manufacturing processes) there is only very little work documented in the literature (Heumann, 1983), (Williams, 1983). Most authors focus on geometric changes (Hamraz *et al.*, 2012).

Conclusion 1:

Components of a product model are designed and changed by the action of multiple disciplines (Clarkson *et al.*, 2004). An appropriate data model has to support the combination of changes on the discipline-specific elements of the product model. Therefore it has to support the ability to derive different view specific to the particular disciplines and their elements. Additional requirements are the reusability of components, transparency concerning that reuse, and patency of the data model (respectively the state changes of the individual components) from early stages of the development process to the end of production.

In summary the complexity regarding the data model arises from the need to map the correct versions of the different information carriers to one another and to manage the dependencies between changes on one carrier to the others. Therefore we propose a novel conceptual view (of components) of virtual product models.

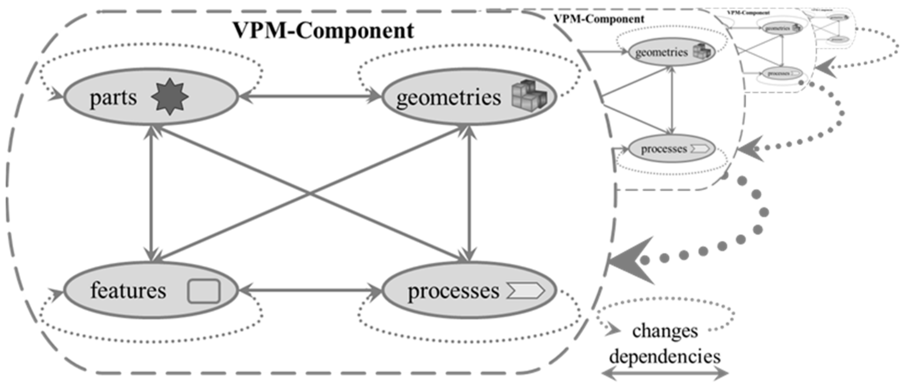

Figure 2. Schematic representation of changes on changes

3.2 Organization

The organization of development projects with the aim to increase the degree of reuse of existing components has great impact on the complexity of the virtual product model. There are different possibilities to model relationships between development partners (brands) and products where the same VPM-Cs are used. In this case reuse can arise in two ways:

- *Centralized* by the instruction of a superior entity
- *Decentralized* during the deployment by using suitable tools for finding and linking existing (or nascent) VPM-Cs

We consider an example to discuss the effects of different configurations. We consider a single VPM-C used in three products (I, II, and III). The corresponding development projects are conducted by two collaboration partners (CP_1 and CP_2) where product I and II belong to CP_1 and product III to CP_2 .

In a *centralized project structure* one of the following settings concerning a superior entity can be distinguished:

- CP_1, responsible for the products I and II
- CP_2, responsible for the product III
- A (mostly) independent third authority (e.g. the corporate management)

In practice, the centralized collaboration setting appears to be the most common. One collaboration partner (= *lead designer*) develops the VPM-C and provides it to the others. Whether the lead designer was elected by choice of all partners, because of historical conditions inside the group, or by a coordinator, does not make any difference in terms of the flow of the collaboration process. In each case only one partner develops the VPM-C while the others are not (or only partially) involved in the development process. Therefore, the VPM-C might not meet all his local requirements and he is probably forced to make changes to his exclusive VPM-Cs (that are not shared with other partners). Therefore he is restricted in its freedom of decision concerning the overall design of his product.

In a *decentralized project structure*, where no lead designer is set, it is much more complex to determine the scope of the shared VPM-Cs. Due to the absence of a lead designer (or coordinator); the reuse of VPM-Cs is not necessarily fixed and therefore not defined at the beginning of the collaboration. Rather the degree of reuse has to increase dynamically over time. Therefore the following methods and tools are required:

- To find VPM-Cs that already exist in a shared repository;
- To alter sections of a VPM-C (e.g. geometrical dimensions) out of a (domain-) specific view on the overall product model (e.g. a CAE-Tool only views the CAD-Files and hides part and process information);
- To define reuse of an existing VPM-C in a new context.

According to the use case described in Section 2.3, the designer is able to increase the degree of reuse by using the methods mentioned above. Because the VPM-C also contains information about procurement channels (part), manufacturing processes (for different factories) and technical descriptions, it can be passed directly to the next process steps. The manufacturing process planner can take over these processes or use them as a basis for further planning at least.

In order to enable increasing the degree of reuse in a decentralized project structure, the following conceptual artefacts are of interest:

- A formal description of the characteristics of the VPM-Cs (particularly on a geometric level) for example based on ontologies as proposed by (Stiefel *et al.*, 2012);
- A formal description of the specification that enables a user to retrieve an VPM-C;
- A shared and searchable repository where VPM-C are stored in consistent states.

Furthermore, a combination of a *centralized project structure* with *decentralized aspects* could be possible. In this case a superior entity would define a set of VPM-Cs that have to be reused as a so called minimum scope while the designers are able to increase the predefined degree of reuse by using the above mentioned dynamic methods.

Conclusion 2:

In cross-brand collaborations (group of companies), the need for coordination and rules that concern the allocation and usage of product model components inevitably arises.

3.3 Process Organization

In most industrial sectors different organizational units with their domain-specific expertise contribute to the creation of a new product. According to (Clarkson *et al.*, 2004), the domain experts have specific views on the overall product model that are just as specialized as their domain-specific expertise (e.g. designers use CAD-Tools such as CATIA or Siemens NX to work on geometric representations of components). (Clarkson *et al.*, 2004) discovered during expert interviews in the context of the development of a platform-based helicopter by GKN Westland, that none of the actors involved in the development process had a detailed view on the entire product. Rather, it was a process of knowledge linking between different development teams.

These developer teams are groups of different domain experts and belong to either (a) the same company respectively an equal partner (multiple brands of the same group) or (b) an external partner. The second case becomes increasingly important because of the decreasing vertical range of manufacture and the resulting integration of collaboration partners (VDA, 2008). These factors affect mainly the formalism and not necessarily the complexity of the change and the integration processes. Therefore, we use the more general term collaboration partner for both alternatives in the following.

Conclusion 3:

In the context of cross-brand collaboration, the processes of data creation and data modification particularly lead to high efforts concerning maintenance of a consistent sequence of state

transition. Different roles change their domain specific parts of the virtual product model component (VPM-C). These changes will most likely lead to changes in other disciplines (and other VPM-Cs including the same disciplines). In order to reach a consistent state concerning these "avalanches", those changes need to be coordinated.
E.g. changes on geometry A lead to changes on the technical description and the manufacturing process and to changes on geometries B and C which in turn lead to changes on their descriptions, processes etc. Taking into account not only the impact of geometric changes, as in (Clarkson *et al.*, 2004) or (Morkos and Summers), but also the other information carriers inside a component, the management and coordination of these cascading effects means a very high effort.

4 TO CORE BUILDING BLOCKS TO THE OPERATIONALIZATION OF CHANGES ON CHANGES

By *changes on changes*, we mean our understanding of the life cycle of a VPM-C that is described in Section 3.1 and illustrated in Figure 2. In order to deal with *changes on changes*, we propose an integrated data model based on feature models similar to (Klawitter and Rock, 2013) and an agent-based approach for the coordination of changes. These two aspects will be discussed in the next two subsections.

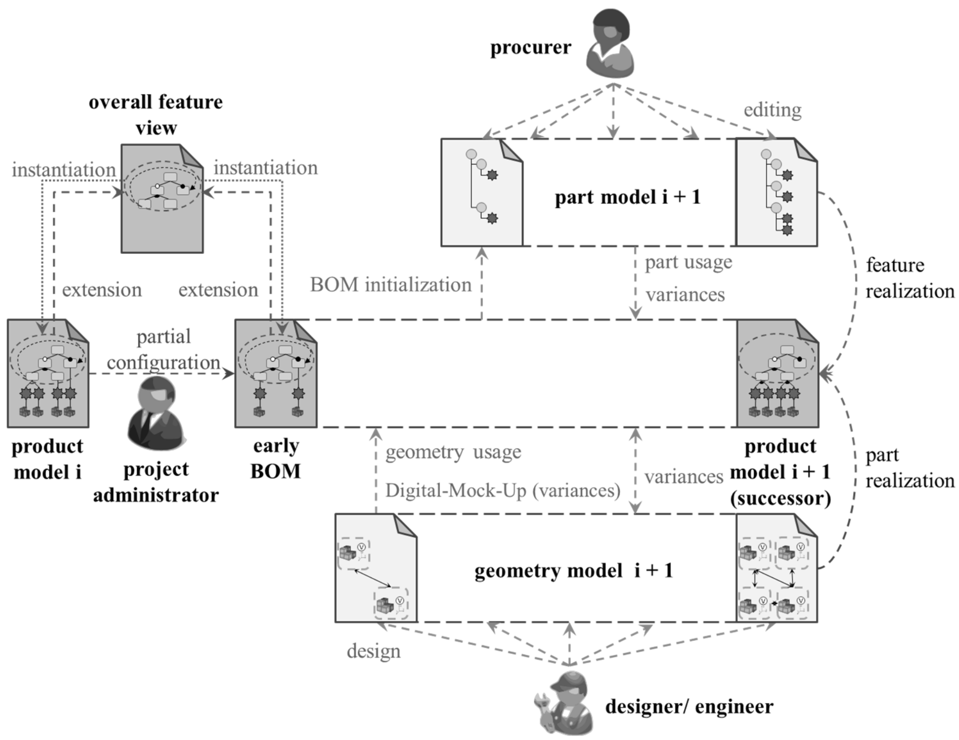

Figure 3. Integrated product model that is extended from domain-specific views

4.1 Integrated Data Model

To keep track of all change, the corresponding versions of each element type and the relationships between the correct versions of these types of the virtual product model need to be in consistent states. Considering the whole product development process as a sequence of changes on domain-specific fragments (of the product), these changes need to be tracked from the early phases to the end of production. Therefore, we propose a cross-organisational virtual product model that "develops" along the product development process. In contrast to most other work our VPM is not only integrated but expanded by different domain experts in varying phases of the product development process. Furthermore we pay attention to changes that arise from different departments and that have to be coordinated in order to keep the VPM (respectively its components) in a consistent state. Due to the multitude of intra and inter component dependencies, domain-specific views on the VPM have to be provided. These views should present the elements indispensable from the perspectives of the

respective experts (see Section 2.5). Figure 3 illustrates the integrated VPM we derived from a predecessor model and which is extended from domain-specific views.
We propose an approach based on feature models for the early BOM and the VPM. Feature models emerged from the field of software product line engineering. The basic idea is to model "a set of related products that share more commonalities than variabilities" (Benavides *et al.*, 2010), which in turn is very familiar to the concepts presented in this paper. A feature represents a category of the VPM-C subordinated to it, e.g. each VPM-C representing a rim would be subordinated to the feature *rim* which in turn would be subordinated to a feature called *wheel* as rims (and tires) are subparts of wheels. Furthermore there field of feature models (FM) and especially the automated analysis of feature models are widely spread and well understood in the literature. For example (Benavides *et al.*, 2010) provide a detailed overview of the knowledge gained in the last two decades. There are several automated analyses on FMs (e.g. determination of *core feature* – features that occur in every product represented by the FM) that could contribute to transparency regarding the usage of components in multiple products of the same product line (modelled in one FM). Therefore the feature rim would be expanded by the corresponding VPM-C. In order to use this analysis not only on a single product line but on all of them, we propose an overall categorization of VPM-C (based on FMs) that is used as a basis for all product models. This hierarchical categorization framework and the mapping from VPM-Cs provide the following abilities:

- Finding similar components (each VPM-C representing a rim is subordinated to the feature *rim*)
- Transparency regarding the reuse of VPM-C in different products of the same product line or across multiple product lines because of the hierarchical categorization
- Definition of the early BOM (as the base for the development of the next product generation) by partial configuration of an existing feature based product model

The left side of Figure 3 illustrates this concept.

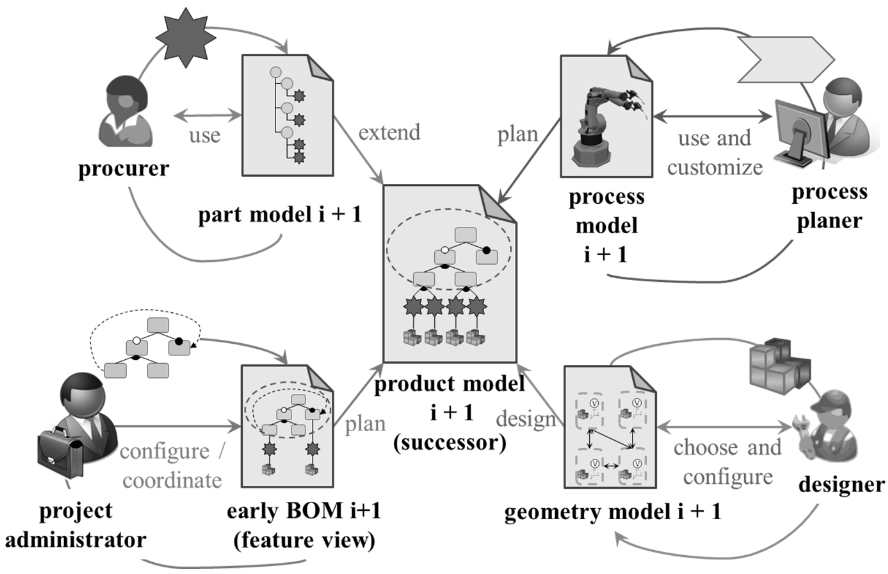

Figure 4. Extension of the integrated product model by coordinating changes on domain-specific elements

4.2 Agents for coordinating changes on changes

As mentioned above and illustrated in Figure 4, the elements of a VPM-C are created and maintained by domain experts that work on specialized views (e.g. a geometry model for designers and a part model for purchasers). The combination of these elements (that arise in those specialized views) is a very challenging task. As described in Section 3, the information overload an individual would have to handle in order to manage the dependencies between the versions of each element is overwhelming. In practice managing the dependencies is done in regular iterations by committees, which, in turn, is a very time-consuming procedure. By considering the reuse of VPM-Cs (and not only their elements) this coordination process can be (partly) automated, because the dependencies between those elements are reused as well. Therefore our approach is based on *active components* where each VPM-C (or its elements, respectively) owns specific knowledge and can act (partially) autonomously.

Example: A designer is looking for a rim to reuse in a new project. Therefore, the designer takes one that realizes the *feature rim* (see Section 4.1) and that meets the requirements. Because this rim was previously used (in other projects), it *knows* about its assembly processes and proposes them to the process planner. The process planner knows that the necessary tools to assemble the rim are not available at plant A (only bigger nuts where used before). Therefore he has to alter the assembly process or to submit a change request on the rim which leads to an alteration of the geometric representation of the rim (bigger boreholes).
These issues may have one of the following solutions:

- The change request is accepted and the proposed assembly process can be used.
- The change request is rejected because of the impact being to extensive (see *avalanche of change* in the introduction or in (Fricke *et al.*, 2000)).
- The geometry will be revised and the altered assembly process will be related to this new version

The involved actors only react on request concerning their domain. All underlying communication is done by the active component(s). In our approach each component is controlled by a *software agent*. The term *agent* arose from the field of artificial intelligence and is defined as "a software system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives" (Wooldridge, 2009). Agent-based approaches have already been applied successfully to describe loosely coupled collaboration processes among autonomous entities. (Wagner, 2008) described a similar approach for information-technical support in order to reduce manual efforts in the engineering of industrial plants.

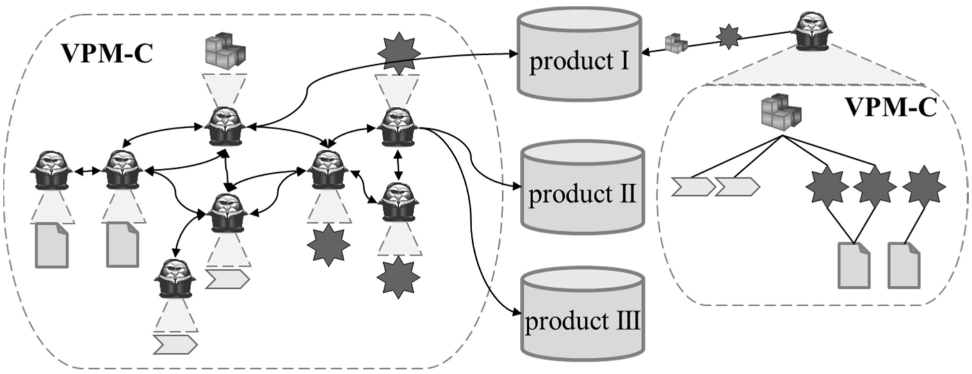

Figure 5. Overview of different agent scenarios

Currently we are investigating alternative scenarios differing in the number of agents and the tasks specific to each agent (see Figure 5 for an exemplary overview). On the one hand, an agent could control each element of a VPM-C (part, geometry, etc.) with knowledge limited to this element and its usages. Therefore an agent would know which products use his element and which elements are related to it (in the same or a different VPM-C). On the other hand, one agent could control a VPM-C containing all its elements. Therefore, this agent has to manage all the relations within his VPM-C, versions of the elements used in different products, and dependencies to other VPM-C.

5 CONCLUSION AND FUTURE WORK

In this concept paper, we performed an analysis of challenges in managing the development of complex products, where several domain experts take part in the development of new products and each domain has its own and specialized view on an overall product model. Consequently, we argued that a **virtual product model** (VPM) has to support **reusability**, **transparency**, and **patency**. Especially in cross-brand collaborations, the administration of this VPM is not manageable without suitable methods and tools.
In order to decrease the amount of information each actor has to manage, we suggested modelling each component of a virtual product model (VPM-C) as a combination of **parts**, **geometries**, **processes** and **features**. These elements are managed within domain-specific views. Each consistent state of the relations between the correct versions of these elements establishes a new version of the VPM-C. For a VPM supporting these **changes on changes** we illustrated our concept based on **feature models**. The VPM is **extended** by domain experts using their **specialized views** across the process. A supporting information system has to deal with these *changes on changes* in order to make the

complexity manageable. To handle the dependencies between elements of a VPM-C we suggested a concept for an agent-based approach.
To summarize, the main contributions of this paper are: (1) a novel conceptual view of virtual product models derived from industrial use case scenarios and (2) an outline of a possible realization of VPM-Cs and of an agent-based *changes on changes* workflow architecture. We believe that these contributions are valuable even in the absence of hard experimental or empirical evaluation results.
Future research will aim at: (1) the elaboration of the different agent scenarios in order to determine suitable modelling requirements and abstractions; (2) the exemplary mapping of a product line in the VPM (based on feature models); and (3) a prototypical implementation to support the use cases from Section 2 taking into account the results from (1) and (2). In particular, this implementation should take industrial software standards (such as Siemens TeamCenter) and resulting integration requirements into account.

REFERENCES

Benavides, D., Segura, S. and Ruiz-Cortés, A. (2010), "Automated Analysis of Feature Models 20 Years Later: A Literature Review", *Inf. Syst.*, Vol. 35 No. 6, pp. 615–636.

Bucciarelli, L.L. (1994), *Designing Engineers, Inside technology*, INSTITUTE OF TECHNOLOGY.

Cheng, H. and Chu, X. (2012), "A network-based assessment approach for change impacts on complex product", *Journal of Intelligent Manufacturing*, Vol. 23 No. 4, pp. 1419–1431.

Clark, K.B. and Fujimoto, T. (1991), *Product development performance: Strategy, organization, and management in the world auto industry*, Harvard Business School Press, Boston, Mass.

Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting Change Propagation in Complex Design", *Journal of Mechanical Design*, Vol. 126 No. 5, pp. 788–798.

Cross, N. (2000), *Engineering design methods: strategies for product design*, Wiley.

Fricke, E., Gebhard, B., Negele, H. and Igenbergs, E. (2000), "Coping with changes: Causes, findings, and strategies", *Systems Engineering*, Vol. 3 No. 4, pp. 169–179.

Hamraz, B., Caldwell, N. H. M. and Clarkson, P.J. (2012), "A Holistic Categorization Framework for Literature on Engineering Change Management", *Systems Engineering*, Vol. 16 No. 4, pp. 473–505.

Heumann, A.K. (1983), "Master production planning and the engineerd to order product", in *Proceeding of the 26th Annual International Conference*, American Production and Inventory Control Society, pp. 293–296.

Jarrat, T. A. W., Eckert, C.M. and Clarkson, P.J. (2004), "Engineering Change", in Eckert, C.M. and Clarkson, P.J. (Eds.), *Design process improvement*, Springer, New York, pp. 262–285.

Jarratt, T., Eckert, C.M., Caldwell, N. and Clarkson, P.J. (2011), "Engineering change: an overview and perspective on the literature", *Research in Engineering Design*, Vol. 22 No. 2, pp. 103–124.

Klawitter, S. and Rock, G. (2013), "Handling of Product Variety Throughout the Product Life-Cycle", in Stjepandić, J., Rock, G. and Bil, C. (Eds.), *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment*, Springer London, pp. 169–179.

McMahon, C.A. (1994), "Observations on Modes of Incremental Change in Design", *Journal of Engineering Design*, Vol. 5 No. 3, pp. 195–209.

Morkos, B. and Summers, J.D., "Requirement Change Propagation Prediction Approach: Results From an Industry Case Study", in *Proceedings of ASME 2010 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Volume 1: 36th Design Automation Conference, Parts A and B, pp. 111–121.

Simon, H.A. (1996), *The sciences of the artificial,* 3. ed., MIT Press, Cambridge, Mass.

Stiefel, P., Hausknecht, C. and Müller, J. (2012), "Using Ontologies to Support Decentral Product Development Processes", in Fischer, K., Müller, J. and Levy, R. (Eds.), *Agent-Based Technologies and Applications for Enterprise Interoperability*, *LNBIP Series*, Vol. 98, Springer Berlin Heidelberg, pp. 114–129.

Tseng, M.M. and Jiao, J. (2007), "Mass Customization. 25", in *Handbook of Industrial Engineering*, John Wiley & Sons, Inc, pp. 684–709.

VDA (2008), *Auto Jahresbericht 2008*, Frankfurt.

VDA (2012), *Jahresbericht 2012*, Frankfurt.

Wagner, T. (2008), "Agentenunterstütztes Engineering von Automatisierungsanlagen", *atp (Automatisierungstechnische Praxis)*, Vol. 104 No. 4, pp. 68–75.

Williams, O.J. (1983), "A case study", in *Proceeding of the 26th Annual International Conference*, American Production and Inventory Control Society, pp. 55–62.

Winterkorn, M. (2009), *Redemanusktipt anlässlich der Jahrespressekonferenz und Investorenkonferenz am 12. März 2009*.

Wooldridge, M. (2009), *An Introduction to MultiAgent Systems,* 2nd ed., John Wiley & Sons.