3 BRIDGING TDM AND PDM SOLUTIONS USING JT IN PLM ARCHITECTURES

Michael Bitzer[*,a], **Martin Eigner**[**,b] **and Florian Gerhardt**[†,c]

Institute for Virtual Product Engineering, University of Kaiserslautern, Kaiserslautern, Germany,
[*]Tel: +49-(0)6312870, [**]Tel: +49-(0)6313873, [†]Tel: +49-(0)6313686, Fax: +49-(0)6313872.
E-mail: [a]bitzer@mv.uni-kl.de, [b]eigner@mv.uni-kl.de, [c]gerhardt@mv.uni-kl.de

In the scope of ever growing complexity amongst engineering networks and products, application of Product-Lifecycle-Management (PLM) concepts has become an essential element in strategic enterprise orientation. The definition and administration of respective system landscapes is complicated by the diversity of predominating Team Data Management (TDM) and Product Data Management (PDM) solutions, whose data is to be held consistently at all times. This diversity, often resulting from product variance and distributed design processes, is especially observed in the automotive industry. In this context, this paper presents a concept for integrating neutral JT (Jupiter Tessellation) based collaboration technologies, in order to bridge information between authoring-system-oriented TDM solutions and comprehensive PDM solutions in a PLM Architecture (PLMA). The benefits of a JT-based data management on the integrative layer are elaborated. Cost and IT system reduction as a result of applying the neutral technology in Computer-Aided Design (CAD) dispensable processes are presented in more detail. Further, the effects of JT integration into up-front PLMA determination approaches are exemplified in terms of a process-oriented approach.

Keywords: Product Lifecycle Management (PLM), PLM Architecture, Jupiter Tessellation (JT), Neutral Data Formats.

1. INTRODUCTION

In complex engineering networks, communication and transparent interpretation of accumulating product data can only be handled by thoroughly reasoned Product-Lifecycle-Management (PLM). PLM incorporates cross-domain and product-oriented management of data and information, including planning, controlling and organizing processes associated with generating and governing this information. Specifically, this counts for the automotive industry. Today, a car of the premium segment consists of thousands of parts and components.[1] During the design phase, each of these parts and components must be stored and handled in a wide range of versions, typically describing different stages in the development process. Design may be characterized by multiple departments, distributed in location, working in different time zones and bound to partnerships, both with Original Equipment Manufacturers (OEMs) and various suppliers.

In this network, an enterprise is coined by diversity of IT-Systems, being the result of political or historical nature, or simply the fact, that each system handles certain functionalities better than others. Any or all of these reasons contribute to a heterogeneous environment, in which product data must be shared and exchanged, both internally and externally. This system environment is one part of a company's Enterprise Architecture (EA). A further subset of the environment this paper calls the PLM Architecture (PLMA), referring to systems associated with PLM.

Figure 1 sketchily depicts a 4-layered PLMA, as typically found in the automotive industry (compare Ref. 2). The first layer includes the sum of authoring systems themselves, such as Computer Aided Design (CAD), Computer Aided Engineering (CAE) or Computer Aided Manufacturing (CAM) systems. Team Data Management (TDM) systems are positioned in the second layer, and serve their purpose in gathering and locally managing data coming from specific authoring systems. Almost all

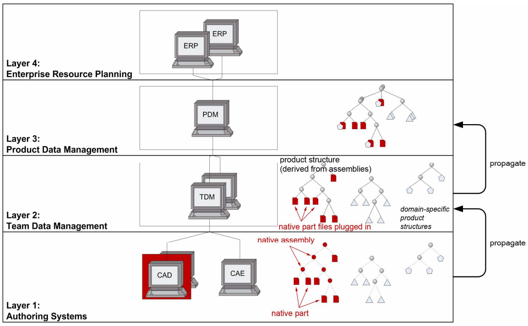

Figure 1. Typical 4-layered PLM Architecture.

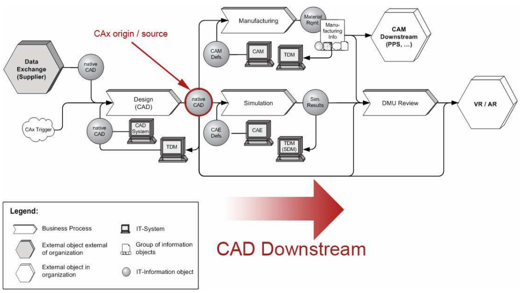

Figure 2. Typical CAx process chain (created with Unity AG-OMEGA Template).

major CAD providers offer a TDM system as well.[3] Product structure information, including CAD data managed in the TDM layer is propagated upwards, and settled into structures positioned in a Product Management System (PDM) on layer 3, which stores and distributes product and process related data across several engineering domains (e.g. mechanic, or electronic), and offers far more integrations than locally running TDM, e.g. providing interfaces into the fourth layer. Here, Enterprise Resource Planning (ERP) supports, amongst others, management of logistical, financial, controlling and human resources.

1.1. Motivation

In a typical 4-layered PLMA, the initial information source for a large subset of all development sub-processes is defined by native CAD systems (Figure 1, red). This is specifically true for processes associated with the first layer. Figure 2 illustrates a simplified, yet predominating CAx process chain, point of origin being the design phase. Here, the product is designed using preferably a single CAD system. Then based on native geometrical and structural data coming from the design phase, CAE, CAM, and Digital Mock-Up (DMU) describe typical downstream CAx processes, followed by integration of DMU and CAD data into Virtual and Augmented Reality (VR/AR) scenarios. These downstream processes rely on native CAD data. Design data coming from external suppliers is for the most part exchanged on the basis of native CAD as well.

Mainly due to the fact that downstream processes today rely on native CAD, respective information coming from both layers 1 and 2 is propagated upwards, as was illustrated in Figure 1. Both, native CAD being the "master format" on layer 1, and native CAD being integrated into product structures on layer 3, lead to a diversity of overhead, as discussed in more detail in Section 4. This overhead describes the fundamental origin of the work presented in this paper. The motivation and issues presented are, in terms of significance, drawn from ongoing studies and initiatives launched by two major boards formed by multiple representatives of the international automotive industry, in which the corresponding author is included and continuously contributing as well.

1.2. Goals

This paper introduces an approach of integrating neutral collaboration technologies into product development, specifically based on the JT (Jupiter Tessellation) data format, in order to keep CAD data dispensable processes free from native information. The technical possibility in realizing this change of paradigm that replaces native CAD with JT being "master format" in development is elaborated, based on typically governing industrial processes, e.g. data exchange and CAx downstream. In doing so, this paper addresses the resulting decrease of financial and organizational overhead that could be achieved by enforcing the presented concept.

Further, this work advises to take into consideration the effects of such a change of paradigm on the determination or redefinition of PLM Architectures (see Subsection 5.2). Exemplary thoughts are illustrated, based on a process-oriented approach thereof. In general, this paper is intended to represent an abstract basis of industry-related research on the potentials of applying neutral data formats like JT in product development, exceeding their primary cause, namely visualization. A more detailed analysis and treatment of this topic is subject to ongoing research.

2. RELATED WORK

JT is found in isolated applications, but there exists no academic research work regarding its integration into a PLMA as a whole. For a detailed insight into the contents of a JT file, see Section 3 and the work presented in.Ref. 4 Further, a general overview in terms of a JT file's setup is given in.Ref. 5

This paper illustrates a concept and the benefits of integrating JT as a neutral collaboration technology from a high level point of view. Detailed mechanisms used to realize engineering collaboration techniques are not discussed, but related to the content of this work, e.g. for sharing streamable JT information between various participants in a change and release management process. Specifically regarding streaming of design data is subject to collaborative CAD, and dealt with, for example, in Refs. 6 and 7.

Stark[8] provides a comprehensive overview on the importance of PLM from a business, as well as a technical point of view. PLM is not a system. Much rather, it is the activity of managing a company's products all the way across their lifecycles in the most effective way,[8] meaning it's a concept. Accordingly, there exist various methods for introducing and maintaining PLM as part of the Enterprise Architecture. VDI 2219[9] presents a well known guideline in this context. Describing amongst others an "as-is", a "to-be", and a "system-selection" phase, it is closely correlated to the PPA (**P**rocess oriented determination of **P**LM **A**rchitectures) approach considered in this paper, and in its roots described in Section 5.2. For detailed information, please refer to. Refs. 10-13.

The term Enterprise Architecture is commonly used in a broader sense, integrating a diversity of architectural domains. Respectively, Jonkers *et al.*[14] give a good insight and overview on this subject. Correlated, the IEEE Standard 1471–2000[15] describes architecture as being "*the fundamental organisation of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution*". The focus of this paper lies on a typical 4-layered PLM subset of EA components and relations, newly considering the possibility and impact of a continuous JT integration thereon.

3. BACKGROUND ON THE JT DATA FORMAT

JT is, from an industrial point of view, a widely accepted, system-neutral file format, that was developed by Unigraphics Solutions (now Siemens PLM Software). It is referred to as a de-facto industrial standard, currently in the process of being officially standardized by the International Organization for Standardization (ISO) as well. While focused on visualization, the corresponding author works together with the two boards mentioned in Section 1.1 to further establish JT especially in the CAx process chain, deriving various requirements that shall serve as input for the ongoing standardization process.

JT holds a variety of information that can be structured into:

- product structure (including visual attributes)
- metadata
- exact geometrical data (primitives, B-Rep)
- tessellated geometrical data (at multiple levels of detail)

While the reduction of overhead in CAx conversions is also one of the primary goals of the Standard for the Exchange of Product Model Data (STEP), it has not prevailed in being established as a fully fledged process-supporting data format, specifically regarding processes that follow design. JT provides major benefits. It is widely accepted, in part due to the publicly available viewer. It is a binary format, in which information can be highly compressed using various CODECs, allowing JT files to remain lightweight. In addition, data is stored in segments or distributed over multiple physical files. In comparison to having to manipulate or holistically parse one file, contents can be altered in target and streamed dynamically. Other benefits are the transparent documentation in terms of the available file format reference, and the fact that the format provides the possibility to directly store GPU shaders, suiting it amongst others for photorealistic VR applications.

4. ENFORCING JT AS A NEUTRAL "MASTER FORMAT" TECHNOLOGY

As mentioned in Section 3, JT harbors containers for managing structural and geometrical information, tessellated and exact. Therefore, JT is primarily used for visualizing parts and assemblies, allowing application of DMU functionality, e.g. collision detection.

JT's possibility to further manage a variety of metadata makes it an interesting candidate for data exchange scenarios, and even more interesting, a candidate to relieve native CAD from being the "master format" for product development sub-processes. Based thereon, integration into PDM product structures becomes a reasonable proceeding, as depicted in Figure 3, and illustrated in the next subsections.

Subsection 4.1 describes the prospects of enforcing JT for the exchange of product data, specifically regarding external exchange scenarios between OEM and supplier. The driving force in this matter is a set of pain points extracted from various suppliers.

First, this paper addresses the advantages of an exchange based on JT, rather than on native data. In order to realize the change of paradigm, integration of JT into the design model as a whole must be assured, suggestively by means of hybrid display technologies, as described in Subsection 4.2. Additionally, JT interfaces must and *can* be provided for downstream processes, such as CAE, CAM and DMU; see Subsection 4.3. With these prerequisites given, JT contents can be used in the CAx process chain, and hence be propagated into PDM product structures on level 3 of the PLMA depicted in Figure 1. Consequences of such an approach are discussed in Subsection 4.4.

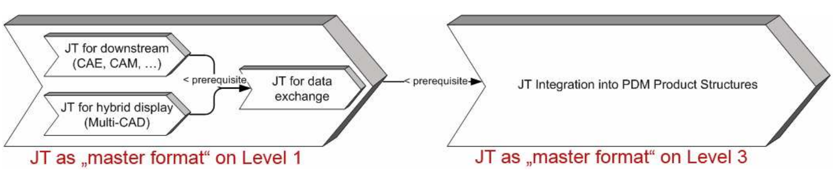

Figure 3. Interdependency of JT enforcements.

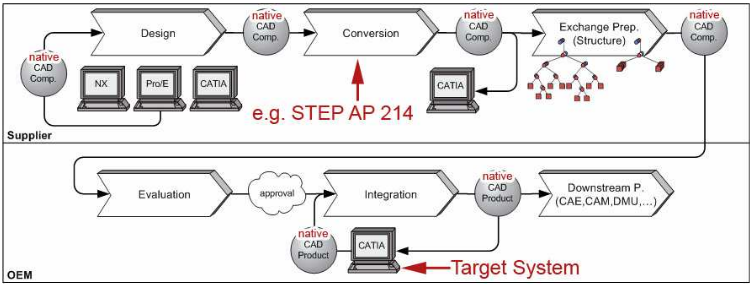

Figure 4. Typical exchange, based on native CAD (created with Unity AG-OMEGA Template).

4.1. JT for Product Data Exchange

Product development being distributed in location and department today, data is constantly communicated as part of the whole, as depicted in Figure 4. For a single exchange scenario, a supplier is typically bound to a target system, provided by the respective OEM.

Since usually contracted by multiple OEMs, a supplier delivers for various target systems. When design does not take place in the system regarding the exchange scenario, the component (CAD Comp.) is converted before transmission, for example using STEP Application Protocols. Depending on constraints (e.g. concerning the depth of the product structure) and quality driven rules (e.g. smoothness of surfaces), the target system data is prepared for exchange (e.g. adjusting the structure of the component assembly). After a positive evaluation on OEM side, the native component data is integrated into the holistic native product data (CAD Product). Such a scenario goes with following pain points (PP):

PP1. Conversion into the target system is associated with time issues.
PP2. Conversion into the target system causes redundantly managed data.
PP3. Conversion into the target system is error prone, converted data is likely to undergo manual correction.
PP4. In an exchange based on native CAD data, means to protect Intellectual Property must be taken.
PP5. The exchange of product data based on native CAD causes tremendous load, often exceeding hundreds of megabytes, even gigabytes.

An exchange scenario based on the JT format would evade the majority of the mentioned problems. Given high quality processors, JT export and transmission avoids having to convert into the target system, sparing time and redundant data (PP1 and PP2). In terms of quality checks, PP3 is certainly a topic yet to be considered in more detail. PPs 4 and 5 are dodged by JT being a lightweight, heavily compressed format, not including parametric and design history. An ideal development stipulates that externally provided product data must not be further processed by the OEM in terms of design. Hence, with a JT file holding specifically structural and geometrical information, the exchange based on this very format must suffice. Prerequisite to enforcing this concept, however, is an integration of JT into CAD systems on the market today, and enabling input interfaces for downstream processes. Both topics are presented in Subsections 4.2 and 4.3.

4.2. JT for Multi-CAD Using Hybrid Display Technologies

Multi-CAD describes the application of a diversity of CAD systems during product development, and their interoperability in terms of integrating product data originating from one CAD system into another. By integration, this paper understands the structural and visual integration only, not the possibility to further manipulate the model. This kind of integration should suffice in an ideal development, initiating change and release request processes for the case of wanting to change the design. In this case, the original authoring system is to be reverted to.

When not considering design history and parametrical data, which is the case in the definition above, integration between two CAD systems can be realized by providing input interfaces for neutral data formats that hold the necessary structural and geometrical data, and means to display this data parallel to existing native CAD data. This technology is described by the term *hybrid display*. It is this technology that also enables the display of large assemblies, showing less relevant geometry in lower resolutions. Additionally, it is possible to display data that is only needed in order to design in its context.

JT holds, amongst others, structural and geometrical data. With the file format originating from Unigraphics Solutions, its already existing, seamless integration into Siemens PLM Software products, such as NX, is not surprising. This very integration is also proof of the technical possibility in using this format for realizing a hybrid display.

The first step now, in realizing the exchange of product data based on this very file format, is to enforce JT input interfaces and hybrid display technologies based thereon in competing CAD systems. Only in doing so, Multi-CAD development becomes a realistic and trouble-free option. Due to the fact that there are no technological hurdles in such integrations, as the NX example demonstrates, this topic is solely a matter of strategically accepting the format from an industrial point of view. The ISO standardization will drive this very acceptance.

4.3. JT for Downstream Processes

The second necessity in allowing JT as primary format for data exchange scenarios is its integration in follow-up, downstream processes. From a technological point of view, there are no major hurdles to be overcome. The corresponding author of this paper has exemplified this statement in terms of an industrial study in cooperation with one of Germany's leading enterprises. Specifically structural and geometrical (both tessellated and exact) information is relevant for typical downstream processes, such as CAE and CAM. JT allows association of arbitrary primitive properties to structure nodes, leading to its capability to insert additional data. On an exemplary basis of a Multibody Simulation (MBS), arbitrary joints could be realized as product structure nodes. Respective attributes, such as degrees of freedom, working angle, etc., can be held by key/value pairs using the mentioned property containers. Even though it is technically possible to import a JT file including CAE pre-processing data (such as force application), in an ideal CAx chain only the most necessary information should be stored in the input format, for example including body mass or material data that allows the derivation of mass. The actual definition of the simulation model is to be done in the pre-processor itself.

4.4. Integration of JT into the Comprehensive PDM Layer

Referencing back Figure 3, Subsections 4.1, 4.2 and 4.3 have described the possibility of enforcing JT as the "master format" in terms of a CAx process chain on layer 1 of a 4-layered PLM Architecture. This prerequisite given, this paper now introduces the concept of integrating JT as the driving format in the third layer as well, as illustrated in Figure 5.

With the possibility to derive JT files for each native CAD part (on the first layer), JT data may be plugged into the product structure of layer 2 (parallel to native, wherever native data exists). With the concept for neutral data exchange presented in the earlier sections, native CAD data is not to be found in each node of the product structure. TDM information is propagated into the third layer, where the authors enforce the idea to no longer hold native CAD information at all. Most processes situated on this layer, for example change and release requests that include viewing, redlining and highlighting suffice neutral geometrical and structural information, enhanced by metadata properties included in JT. Synchronization between layers 2 and 3 associates PDM JT files with TDM JT files. Should in any case be need of native CAD data, JT's support for metadata can function as a bridge on the TDM layer to dynamically get a hold thereof, if existent.

The advantages of abandoning native CAD from processes situated on the third layer are obvious. Without the necessity of having the according authoring system installed, a broader sense of engineering collaboration is made possible. In terms of sales and presentation derived on the PDM layer, application of Virtual Reality technologies becomes straight forward for example. JT provides containers for holding various scenegraph data, including texturing, Graphical Processing Unit (GPU) shaders and more. The adoption of this data into the PDM product structure makes an automatic derivation of a VR input file, without manual scene preparation feasible.

5. DISCUSSION AND CONCLUSION

This paper has illustrated a concept for enforcing JT as the "master format" in development, keeping CAD dispensable processes free of native data. In the context of a typical 4-layered PLM Architecture, the TDM layer may function as a bridge, communicating native CAD data to PDM related processes only when necessary. It was shown that neutral formats may be used to reduce financial and organizational overhead, based on pain points provided by various suppliers. While this paper cannot give scientifically relevant results in numbers and figures, citations from leading automotive industry representatives state that "conversions between various CAD-solutions and -versions cause an overhead of 15% in development and design". Data exchange based on the JT file format can avoid this overhead. Subsections 5.1 and 5.2 are committed to presenting further steps the authors are currently taking in the scope of the presented topic.

5.1. Procedure Model for a Continuous JT Enforcement

The presented concept has been depicted primarily from an abstract point of view. The corresponding author is working on defining a mapping between various containers provided by JT to a diversity of functionality supported thereof. In the same scope, the detailed analysis and depiction of business processes related to visualization and exchange is currently subject to ongoing research in close cooperation with industry. Future work will illustrate the linkage between these processes and JT functionality, elaborating the applicability of the file format, as well as requirements necessary for an according integration, in order to then actually begin introducing JT as the "master format". Based on the results gained in terms of JT, a general procedure model for a neutral data format rollout will be defined.

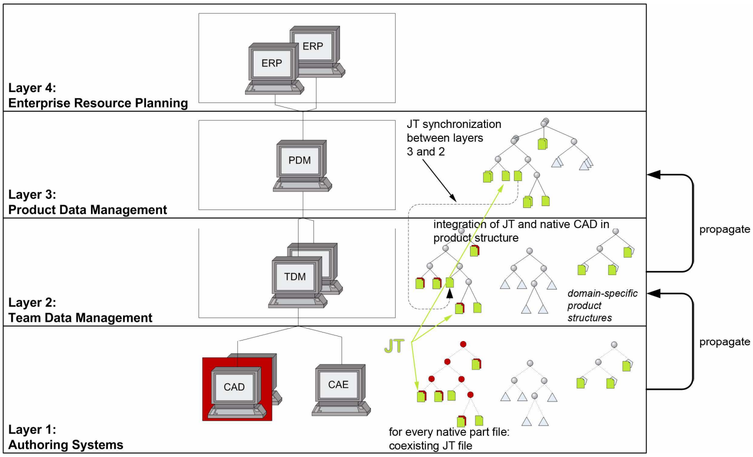

Figure 5. JT integration into a 4-layered PLM Architecture.

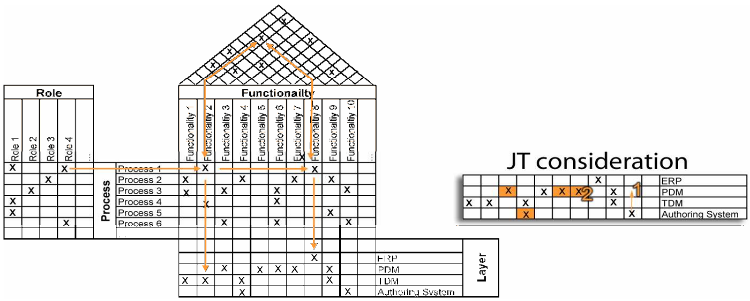

Figure 6. PLM Matrix (left) and JT integration therein (right).

5.2. Effects of a Continuous JT Enforcement on the PPA Approach

The technical capabilities of enforcing JT as the "master format" in product development have a tremendous impact on the determination of PLMA up-front. The introduction or redefinition of PLM as part of an Enterprise Architecture is a sensitive topic as is, both from an industrial point of view, and as a focus of research. Based on the PPA, this Subsection discusses the effects of a continuous JT integration thereon.

PPA describes the approach of a process oriented determination of PLM Architectures. The method aims at selecting from a set of possible PLM solutions based on diversity of information obtained during various phases of the approach, beginning with the analysis of existing business processes inside an enterprise. The approach consists of the three phases *Process analysis and synthesis, Architecture analysis and synthesis* and *System vendor analysis*. The PLM matrix represents the core of the second phase, describing the interrelation between processes, roles, PLM functionalities and the according layers, in which functionalities are to be provided. See Figure 6 (left) for an illustration. Roles are related to processes, which in turn are linked to various PLM functionalities, such as management of change and release, viewing and redlining, etc. On top, functionalities influence each other. Finally, certain functionality can be provided by a system of a certain layer, depicted in the matrix as well.

In a PLM introduction project, given the wish of enforcing JT as the "master format", diversity of functionality that is defined on native CAD today, meaning in the Authoring Systems layer, may shift to be offered in the PDM layer, see Figure 7 (right, 1). JT integration possibilities into various functionalities could be color coded into the matrix, as depicted in Figure 7 (right, 2) as well. This newly introduced information will have an effect on the integrated vendor analysis of the PPA, as not all available systems offer JT support as of today.

REFERENCES

[1] Zagel, M. (2006). Übergreifendes Konzept zur Strukturierung variantenreicher Produkte und Vorgehensweise zur iterative Produktstruktur-Optimierung. *Technische Universität Kaiserslautern*, Kaiserslautern (German).

[2] Burr, H., Vielhaber, M., Deubel, T., Weber, C. and Haasis, S. (2004). CAx/EDM integration: Enabler for methodical benefits in the design process. *Proceedings of the DESIGN 8th International Design Conference (DESIGN 2004)*, 2:833–840.

[3] Krastel, M. (2002). Integration multidisziplinärer Simulations — und Berechnungsmodelle in PDM-Systeme. Shaker Verlag, Aachen, (German).

[4] UGS Corporation (2006). SIEMENS JT File Format Reference Version 8.1 Rev-B. http://www.jtopen.com/technology/format_reference.shtml

[5] Eigner, M. and Gerhardt, F. (2008). Dynamic and Conceptual DMU. *Proceedings of the 10th International Design Conference (DESIGN 2008)*, 1:619–625.

[6] Li, W. D., Lu, W. F., Fuh, J. Y. H. and Wong, Y. S. (2005). Collaborative Computer-Aided Design — research and development status. *Computer-Aided Design*, 37:931–940.

[7] Fuh, J. Y. H and Li, W. D. (2005). Advances in collaborative CAD — the-state-of-the-art. *Computer-Aided Design*, 37:571–581.

[8] Stark, J. (2005). Product Lifecycle Management: Paradigm for 21st Century Product Realisation, Springer.

[9] VDI-Gesellschaft Entwicklung Konstruktion Vertrieb. (2002). VDI 2219: Information Technology in Product Development — Introduction and Economics.

[10] Bitzer, M., Eigner, M. and Vielhaber, M. (2008). Impacts of Design Process Characteristics on the Selection of PLM Architectures. *Proceedings of the 10th International Design Conference (DESIGN 2008)*, 2:901–910.

[11] Bitzer, M., Burr, H. and Eigner, M. (2007). Produktlebenszyklus mit Unterbrechungen — Abstimmungsbedarf zwischen Konstruktion und Produktion. *ZWF — Zeitschrift für wirtschaftlichen Fabrikbetrieb; Heft Nr. 9/2007*, Carl Hanser Verlag, pp. 582–586 (German).

[12] Bitzer, M., Eigner, M. and Langlotz M. (2007). Prozessorientierte PLM-Architektur als methodische Unterstützung bei PLM-Einführungen (Teil 1). eDM Report, Heft Nr. 4/2007, Dressler Verlag, pp. 3–7, (German).

[13] Bitzer, M., Eigner, M. and Vielhaber, M. Ermittlung von betriebsspezifischen PLM-Lösungen, Industrie Management — Zeitschrift für industrielle Geschäftsprozesse; Heft Nr. 3/2008 (German).

[14] Jonkers, H., Lankhorst, M. M., Doest, H. W. L., Arbab, F., Bosma, H. and Wieringa, R. J. (2006). Enterprise architecture: Management tool and blueprint for the organization, *Proceedings of Information Systems Frontiers*, pp. 63–66.

[15] IEEE Computer Society (2000). IEEE Recommended Practive for Architectural Description of Software Intensive Systems. *IEEE Standard* 1471–2000, October 9.