Concept to Enrichen Lightweight, Neutral Data Formats with CAD-based Feature Technology

Martin Eigner[1], Sebastian Handschuh[2] and Florian Gerhardt[3]

[1]University of Kaiserslautern, eigner@mv.uni-kl.de
[2]Daimler AG, sebastian.handschuh@daimler.com
[3]University of Kaiserslautern, gerhardt@mv.uni-kl.de

ABSTRACT

The need of feature information in CAD-systems is undisputed. In order to accelerate the product development process, the use of features should be considered to its full extent. With neutral, lightweight data formats becoming more popular in terms of replacing native CAD-data in many follow up processes in the scope of Product Lifecycle Management (PLM), a consistent use of feature information in such formats must be adapted. By the example of JT, we present a concept to combine neutral, lightweight data formats with a respective XML schema/dataset, enriching the format's application potentials.

Keywords: features, neutral data formats, JT, Product Lifecycle Management.
DOI: 10.3722/cadaps.2010.89-99

1 INTRODUCTION

The rapid change of the surrounding conditions and globalized markets force many companies to adapt their processes to a shorter time to market, reducing costs of development processes and optimize their products in order to fulfill stricter environmental regulations.

1.1 Motivation

One proven possibility to improve a process-chain is to carry on and provide different forms of knowledge for efficient interpretation and reuse. Such knowledge may be given by including feature information within Computer Adied Design (CAD) files that underly the respective processes. The importance of feature information usage comes with the need for:

- reducing time to market
- maintaining external development partnerships
- allocated development
- respecting different production conditions

The very fact that over the years CAD files have established themselves as a knowledge carrier by enhancing product data with feature knowledge may well be the reason that today, multiple processes within today's product development expect native CAD as the primary input source. In turn though, this has led to in part tremendously large data sizes, excessive conversion activities and measures to protect intellectual property regarding sensitive data in data exchange scenarios, all of which are counter-productive issues. In addition to using native CAD files, native CAD systems are being applied as the very resource to perform process-related tasks, leading to high licensing costs, especially compared to applying e.g. more process-specific or viewer-based tools.

Accordingly, neutral leightweight data formats have gained a lot of momentum lately. Much work has been published recently, suggesting a more extensive integration of non-native data into processes that follow design [1],[5],[8].

We propose an enhancement of that paradigm by integrating additional information that relates to features into respective formats, in order not to lose the proven benefits that come with carrying on such knowledge.

1.2 Need for Action

The integration of feature technology into the digital product development process is needed to support active and passive knowledge engineering [18]. Hereby, the passive use of knowledge engineering is understood as knowledge representation (how can knowledge be made accessible?) while active use refers to the handling of knowledge (how can knowledge be processed?). Fig. 1 shows the long history of the feature development, indicating the importance of the topic in terms of knowledge engineering.

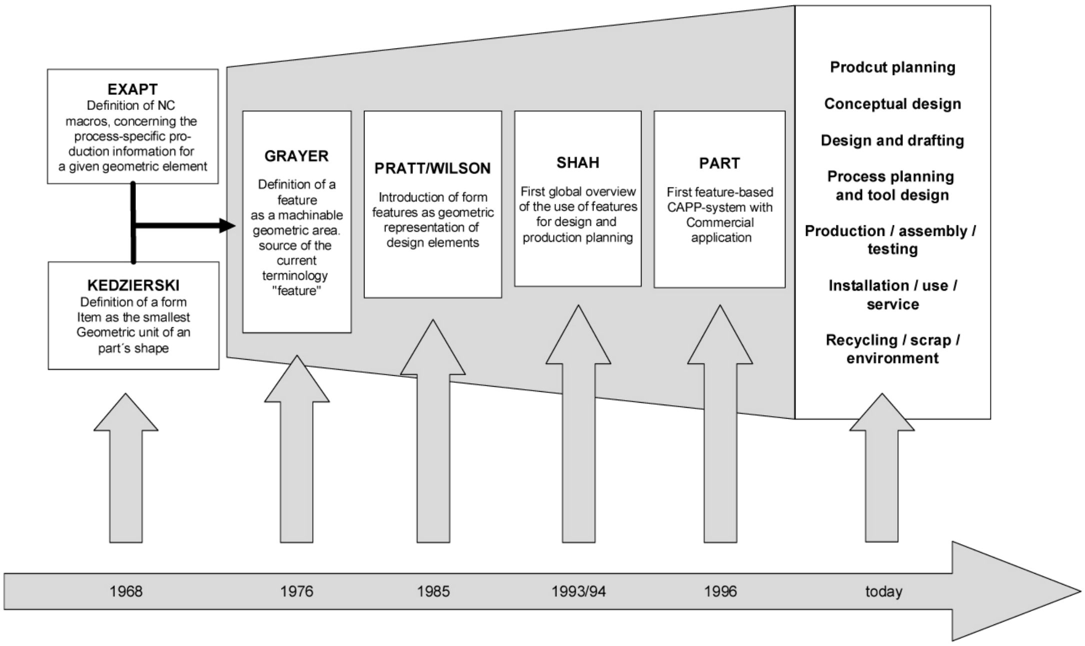

Fig. 1: History of features development, relating to [2],[18].

Working closely with well known industrial partners, as well as the internationally renowned ProSTEP iViP association and Verband der Automobilindustrie (VDA), the Institute for Virtual Product Engineering (VPE) continuously tries to find new ways to enable neutral data formats in follow up processes [8] and make sure that there is an integrated information chain throughout the product development process. From underlying activities, the conclusion can be drawn that many of today's follow up processes could be rearranged to a lightweight, neutral data format like JT (Jupiter Tessellation), but the essential integration of kinematical and feature-related data into such formats remains a yet unsolved challenge.

Much meta-information generated in CAD-systems gets lost during the conversion from CAD to neutral data formats, including such relating to feature technology. There is a need to flexibly "re-enrichen" considered data formats, either by suggesting respective containers, or by combining those that do exist with other means. We propose a new methodology to enhance conversion tools with XML-based meta-data tools, thereby providing the necessary feature information.

2 PRELIMINARIES AND RELATED WORK

Because dealing with the enrichment of lightweight data formats by the example of JT, an overview of the underlying technology and a short comparison to the Standard for the Exchange of Product model data (STEP) is given in the following, prior to addressing the term feature.

JT is a widely accepted, system-neutral file format that was developed by Unigraphics Solutions (now Siemens PLM Software). Being an industrial standard as of today, a JT ISO standardization was successfully triggered by the end of 2009. "At its core, the JT format is a scene graph with CAD-specific node and attribute support" [14]. The most obvious difference to STEP is defined by the slim storage of tessellated geometry, at several levels of detail (LOD). This is traced back to the historical fact that the initial purpose of formats like JT was to be able to simply "view" product representations without the need of a CAD-system, as compared to fully describing product data throughout a product's life cycle. Hence, the term lightweight has evolved. Some of the benefits compared to STEP are: high compression via multiple CODECs, quick access to selective contents because the file format is binary and different segments can be addressed via a table of contents, and a transparent documentation of approximately 300 pages. There exist freely available viewers to visually analyze product data. Summarizing, JT contents can be classified into those depicted in Fig. 2.

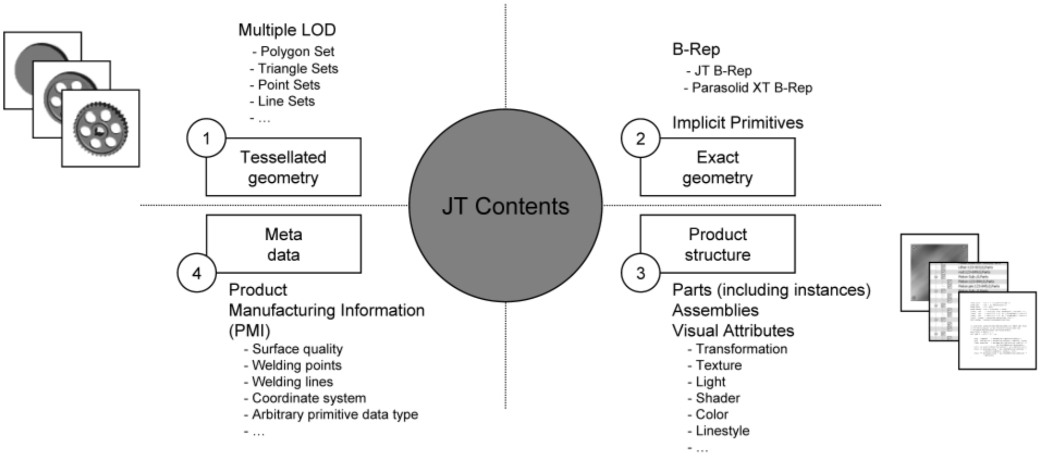

Fig. 2: JT contents.

PLM XML is a comprehensive data format based on XML and is also considered an industrial standard. It has historically evolved from facilitating product lifecycle interoperability [15] in the context of applying several Mechanical CAD (MCAD)-systems in cross-enterprise engineering, and includes support for managed, revision- and view-based product data as in common PDM-systems. Because JT can store contents in shattered files, for example encapsulating geometry, PLM XML enjoys being used in combination with JT, storing product structure and company-internal meta data in PLM XML and externally referencing per-part JT geometry.

Recent works have dealt with enriching JT contents, based on adding virtual nodes within the product structure. Precisely, based on a comprehensive study of various kinematical descriptions existent, we have added kinematical systems in this manner, as elaborated in [6]. It is conceivable to also encapsulate our proposal of the kinematic description in XML-files, similar to the approach presented here for the coupling of feature-related content.

Neither PLM XML nor JT hold explicit containers to carry feature-related information. The feature term is defined in many ways, often depending on the domain and context. Shah et al. state that a “Feature is a chunk of knowledge” [13], which we consider too general. This paper will consecutively use the following feature definition based on the VDI 2218 guideline [18]:

A feature is an aggregation of geometrical elements and/or semantic.

In the above definition, semantic is understood as the interpretation of the meaning of the feature in the different phases of the product development process and the associated information. Based on the work of Vanja et. al. [16], use of feature based systems offer the following benefits:

- documentation and reuse of expert knowledge
- improvement of simultaneous engineering processes and communication
- possibility to check quality of a design draft (early product development phases), in order to compare requirements and functionality to design and producibility
- possibility to compare production process and the costs caused by these process
- possibility to connect product elements

Several dissertations [2],[7],[16] have described the use of feature technology in partly very specialized sub domains and/or in context of using CAD-systems in follow-up processes (e.g. applying knowledge feature technology within deformation technology or the use of assembly features in simulation tools). However, we are not aware of works dealing with providing such technology by using a single lightweight and neutral supporting data format, finding itself situated and supporting processes of these subdomains.

Before addressing our approach to combine JT files, PLMXML and custom XML schema to carry feature-related data, the scope of features considered was defined based on the following classifications.

2.1 Feature Classification by Type

In general features can be divided into two parts, the first being the “type” [16]. The following list illustrates feature types [18] typically found within today’s CAD-systems.

Feature type	*Description*
1. form features	group of geometrical elements, formalized aggregation of form properties in the product model. These features were saved under a common name and could be used if the same geometrical elements were applied several times. Moreover it is possible to connect form features with properties of different property classes
2. semantic features	combination of form-features and semantic which is used in different product development phases
3. implementation features	part of the semantic features for one specific phase of the product development process
4. features for free form surfaces	- Features for design: simplified geometry, used in early CAE-phases, contain mechanical structures - Features for CAD-structure draft: contain free formed surfaces, features could be used to include constraints to contiguous systems - Features for modifying and simplifying of structural drafts as well as overlapping fasteners such as welding spots

Tab. 1: Feature classification by type.

With the exception of type 1, all of the above features include semantics and/or meta-data for processes that follow design. Because form feature usage focuses on the design process itself, this type of feature is not to be considered in our approach.

Furthermore, features that integrate semantics can be classified into different levels of added semantic information as depicted in Fig. 3. Hereby, common Product Manufacturing Information (PMI) defines a very low-level form of feature. The next level represents advanced feature information which will be presented in subsequent. Further levels of added semantic information are conceivable.

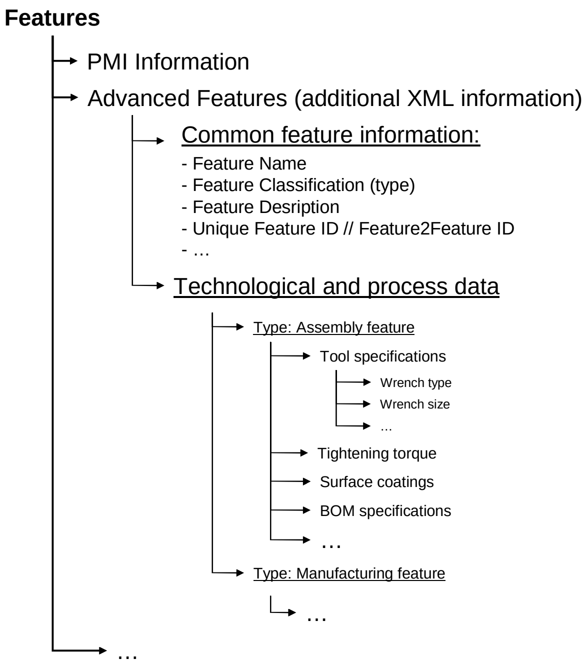

Fig. 3: Further feature classification (extract).

Features may be used repetitively in different domains of product development, exemplary related views depicted in Fig. 4.

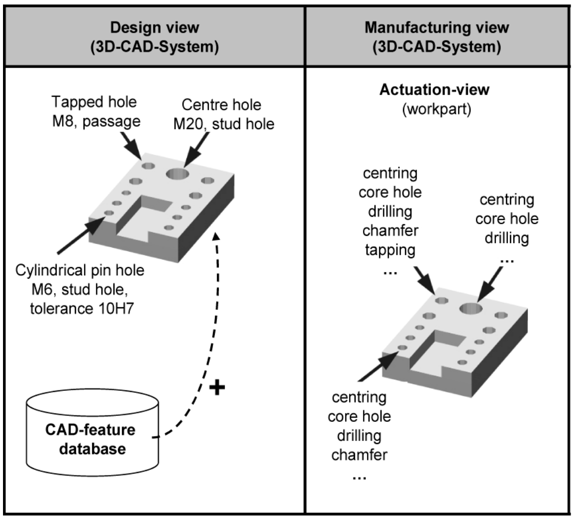

Fig. 4: Exemplary views from different feature usage domains (translated from [18]).

This leads to a further classification: by procedural description. As depicted, all information is added and used in CAD-systems for follow up processes.

2.2 Feature Classification by Procedural Description

With a type being defined (specifically types 2 – 4), a more detailed classification of any feature can be provided by a procedural description [16] regarding its usage scenarios. Based on the work of Vanja [16], we propose a separation into the following feature types:

- design
- measurement
- manufacturing
- quality-management
- assembly
- validation
- cost calculation
- user-defined

Hereby we relate to several publications detailing these very feature types, amongst others the following: [3],[4],[9],[10],[11],[12]. For the concept presented in subsequent, we intend to focus on features used in the context of design, manufacturing, assembly and validation.

3 CONCEPT OF INTEGRATING FEATURE INFORMATION IN JT

Without loss of generality regarding lightweight data formats, we illustrate our concept based on JT, due to our long history relating to this format. We propose using JT to carry feature-related information, dividing the approach into two correlating parts:

1. The features' semantic definitions are to be stored externally, because of the multitude and specialization of data required, depending on context and domain of the feature. An XML schema can be defined in order to have respective XML file instances hold the required content. Fig. 5 gives an insight on the taxonomy, based on the classification mentioned in Fig. 3
2. With XML files holding feature semantics, we suggest enriching the JT format only with information on feature types, and associated geometry (topological information). In addition, JT must further store a reference to the additional XML data. For this proposed enrichment, we suggest to either create specific nodes within the JT product structure, or using the GenericPMI containers or Design groups provided by the data format.

Also, while the JT data format is able to hold almost arbitrary properties, the PLMXML format has evolved to an industrial standard to carry particularly product master data. Hence, we include PLMXML in our approach as well. Both JT and PLMXML can hold information on comprehensive product structures, either case allowing management of feature information on an assembly level.

Fig. 6 depicts a detailed view of the CAD to JT/PLMXML and XML feature translation concept and proposes a data base to store information. In order to attain the two or three different files (XML, JT, and PLMXML) a from CAD, a custom interface is developed based on the CAD-API, basically enhancing the already existent export of JT and PLMXML.

Fig. 6 gives an overview of the information distribuition out of the 3D-CAD source system. It is obvious that not
all information stored in the 3D-CAD-system could be stored in the JT file format such as

- Parametric information
- Construction history
- Technological data

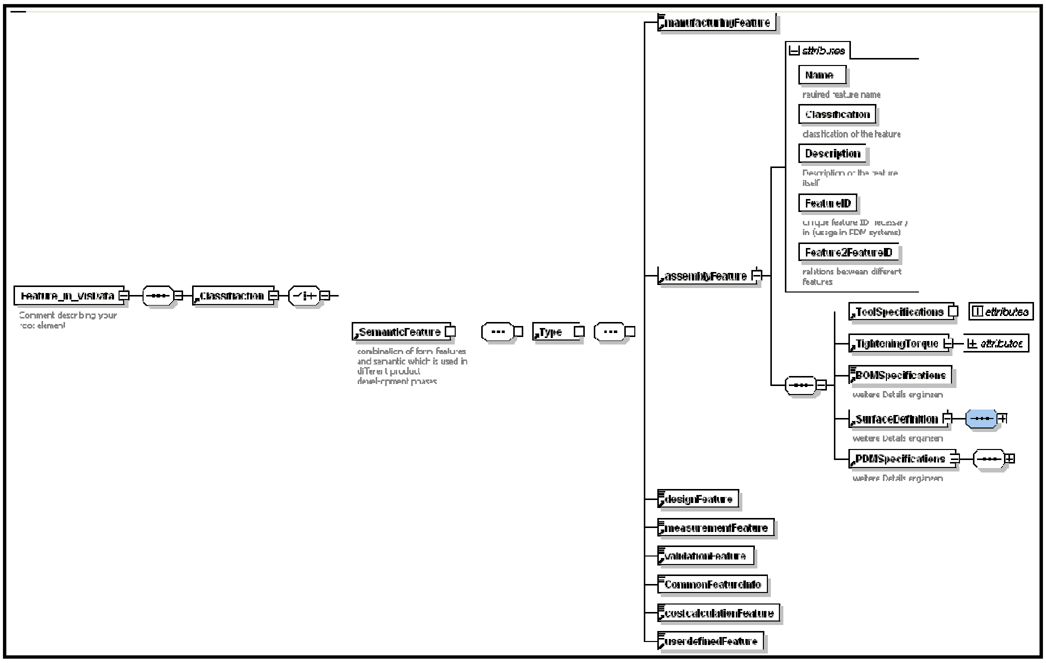

Fig. 5: Feature XML -schema (extract).

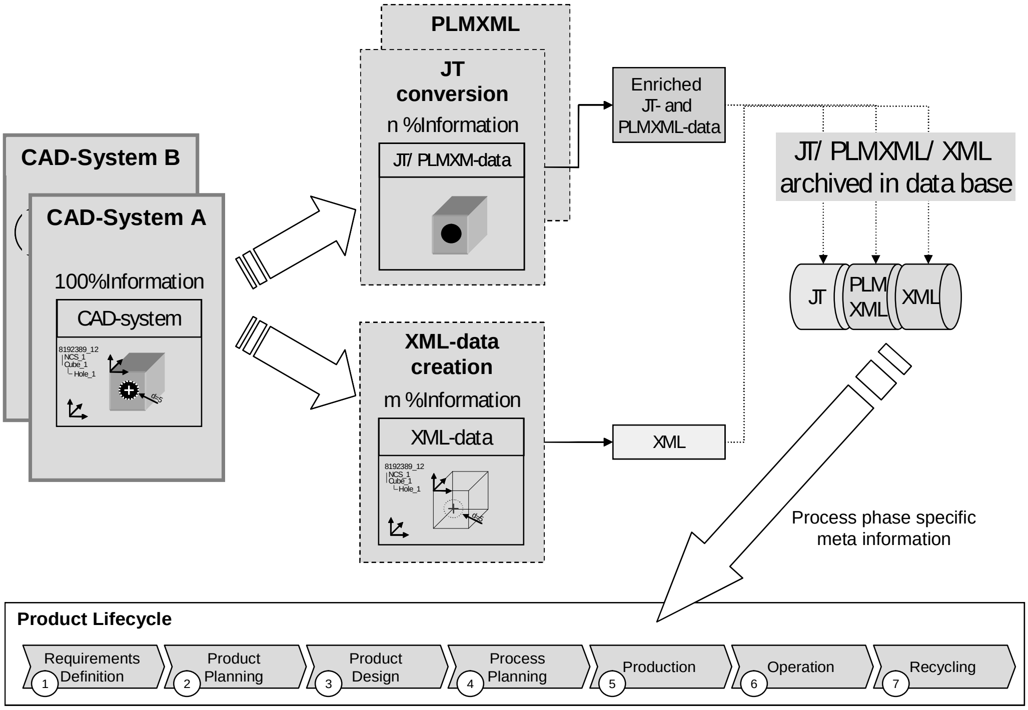

Fig. 6: Detail view of the concept.

Therefore the separation into JT/PLMXML and XML-data is proposed. As mentioned in Section 2.2, a feature can be classified by its procedural description, referring to its usage or stakeholder context. This allows different features relating to the same product data to be utilized within multiple phases of the product lifecylce. Since we imagine different stakeholders to extract feature contents based on a visualized JT, we propose a further classification making use of layer concepts (existent in both PLMXML and JT) already established in Daimler AG product development processes. Specifically for the purpose of filtering product contents, Daimler AG has specified the following nine layers that are already included in PLMXML files:

- Clamping and fixing
- Welding
- Tooling
- Tolerances
- DMU
- Measure Component
- Measure Assembly
- Last revision
- Finished Part

We intend to make use of this layer-based classification to organize defined features.

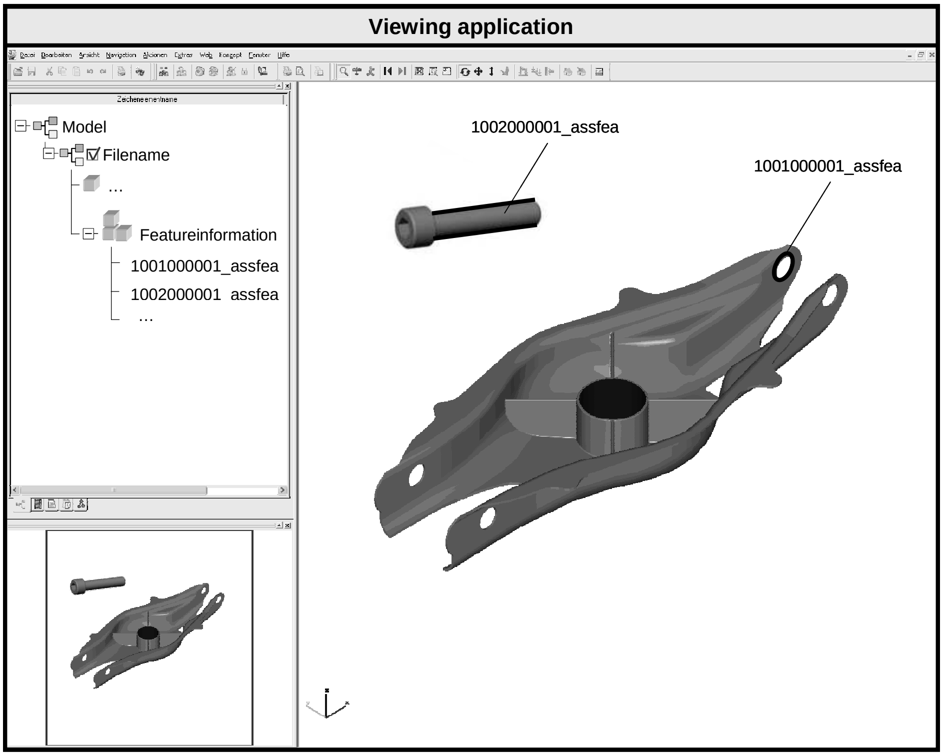

Fig. 7: Feature representation in JT structure.

Fig. **7: Feature representation in JT structure.**

7 shows the enriched JT feature by the example of an assembly feature. Due to fact that feature relations have to be consistent with a feature nomenclature (unique feature ID) is proposed according to DIN 6763 described in the work of Pahl [19].

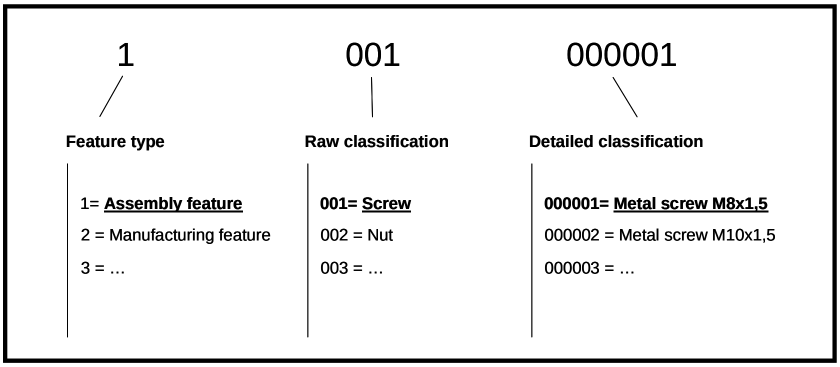

Fig. 8: Feature ID (translated from [19]).

As mentioned, the JT data sets are intended to serve as the driver for feature extraction and representation. We imagine a neutral viewing application, allowing the user to select specific objects that are visualized. With the existing mapping between JT and feature semantics inside the proposed data base, respective meta information can be provided depending on a selected filter. This is illustrated in Fig. 9.

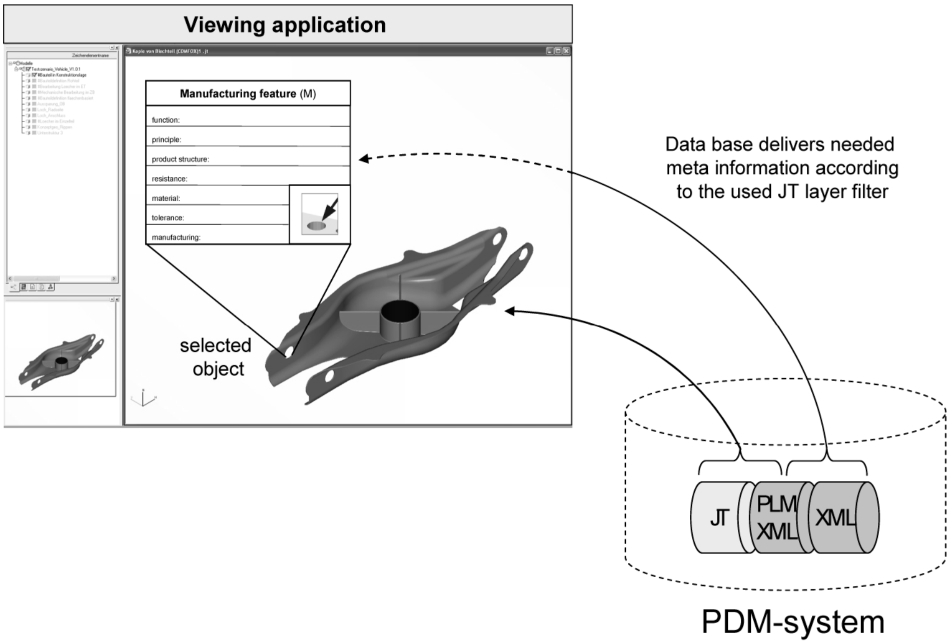

Fig. 9: Concept of feature integration in JT.

Due to its standardized nature and transformation possibilities, we suggest to combine the approach with web technology data formats, such as HTML. Using an Extensible Stylesheet Language Transformation (XSLT), an XHTML page could be generated, displayed in the viewer and also integrated e.g. in subsequent reports.

On a further note: the fact that meta information (XML-based feature definition) and geometry is stored separately, maintains JT's lightweight character and hence, optimized performance.

In order to integrate this concept into automotive product development processes, it is recommended to use a PDM-system to store the datasets. The use of a PDM-system offers the opportunity to store unique feature IDs over the whole product development process. Moreover it is possible to use PDM-functions to realize (version based) positioning of parts (e.g. screws) with transformation matrices.

4 CONCLUSION AND OUTLOOK

In conclusion it can be stated that there is a need to implement/transfer feature information from 3D-CAD-data into neutral data formats, in order to maintain the benefits from both feature-based knowledge transfer and CAD-independent downstream communication. We have presented a concept based on JT in combination with a lightweight web technology (XML) in this paper. One major advantage of our concept is that the JT-file itself stays lean. All process specific meta-information is stored in XML and loaded on-the-fly into the application displaying the JT-dataset or independent systems such as an html-browser. Moreover, it is possible and recommended to store this additional XML-based feature information in a database or PDM-system for example. Using a PDM-system generates additional positive aspects, such as storing unique feature IDs (feature lifecycle management) over the whole product development process or (code based) positioning of features by transformation matrices using PDM-functions. These aspects will be considered in detail in further analyses. Furthermore special feature types such as assembly features need to be implemented by using a database or PDM-system as backbone to ensure their consistency over the whole product development process. The presented concept could be integrated into drawing less manufacturing processes, such as the 3D-Master process at the Daimler AG in future.

When continuously updating the feature information associated with JT files, consistency is a topic that must be addressed as well. When a user modifies a feature in his view, it is going to have to be ensured that the modification doesn't negatively impact the same feature in another view or other features in the same view.

With the proposed concept in mind, the next step will be for us to narrow down and in detail describe the set of features and usage scenarios to further consider. In order to do so, we will document the results (requirements) of a process analysis using an in-house developed methodology, based on previous works in the scope of the activities mentioned in Section 1.2. To identify the demands on neutral data formats and according software tools in more detail (borderlining our features), several business units such as Digital factory, DMU and Simulation were interviewed. As a first result three interdependent magnitudes of influence were identified, as depicted in Fig. 10. Hereby, the product development process is assumed to be the driving constant.

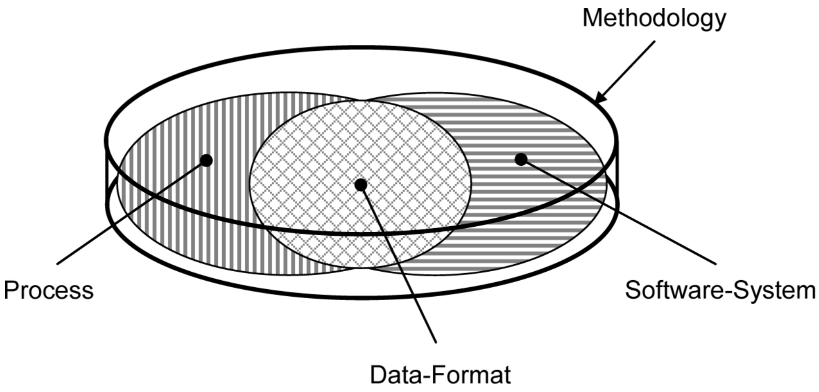

Fig. 10: Process analysis methodology.

In order to provide the XML-based feature semantics, we're developing an XML interface to respective CAD-systems. While doing so, the opportunity has come up to enrichen these interfaces on an even broader term. We're working on a solution to actually rebuild 3D-CAD-system models of interfacing supplier parts by using XML-files instancing a schema we have specified. By developing XML-schema for different CAD-systems we will even be able to support design in context or multi-CAD migration by again using XSLT technology. Combining this approach with the approach presented here, it is also intended to "import" the JT-based geometry, and for selective features enhance it in the CAD-system by subsequently extracting the parameters and feature-information from the XML-schema. This way, not the whole model is rebuilt, much rather only that of interest.

REFERENCES

[1] Bitzer, M.; Eigner, M.; Gerhardt, F.: Bridging TDM and PDM Solutions using JT in PLM Architectures, in: Chakrabarti, A. (Hrsg.): Research into Design - Supporting multiple Facets of Product Development", Bangalore/India, January 7th – 9th, 2009, 207-216.

[2] Bossmann, M.: Feature-basierte Produkt- und Prozessmodelle in der integrierten Produktentstehung, Dissertation Universität des Saarlandes 2007, Schriftenreihe Produktionstechnik, Band 38, LFT, Saarbrücken, 2007.

[3] Ciesla, M.: Feature-basierte Messplanung für Koordinatenmessmaschinen, Dissertation TU Berlin, Berichte aus dem Produktionstechnischen Zentrum Berlin, 1997.

[4] Ehrlenspiel, K.; Schaal, S.: CAD integrierte Kostenkalkulation, Konstruktion 44, 1992, 407–414

[5] Eigner, M.; Gerhardt, F.: Extensible JT Open Tool for Prototypical Process Support, based on OpenSceneGraph and QT, in: IEEE Proceedings of the 2009 13th Inter-national Conference on Computer Supported Cooperative Work in Design, Santiago/Chile, 2009, 734-739.

[6] Eigner, M.; Gerhardt, F.; Gilz: Abbildung von Kinematik in leichtgewichtigen Datenformaten, am Beispiel des ISO/NP PAS 14306 Standards (JT), in: CAD-CAM Engineering Report 1/2010, 56-59.

[7] Franke, C.: Feature-basierte Prozesskettenplanung in der Montage als Basis für die Integration von Simulationswerkzeugen in der Digitalen Fabrik, Dissertation Universität des Saarlandes 2003, Schriftenreihe Produktionstechnik, Band 28, LFT, Saarbrücken, 2003.

[8] Gerhardt, F.: Requirements towards establishing JT in the CAx process chain and in data exchange scenarios, ProSTEP iViP Association, ProSTEP iViP Symposium, Berlin / Germany, May 12th – 13th 2009., Track 10/2

[9] Haasis, S.: Wissens- und featurebasierte Unterstützung der Konstruktion von Stirnradgetrieben unter besonderer Berücksichtigung des Gussgehäuses, Dissertation Universität Stuttgart, Fortschrittberichte VDI, Reihe 1, Nr. 254. VDI-Verlag, Düsseldorf, 1995

[10] Haasis, S.; Ströhle, H.; Karthe, T.; Pfeifle, J.: Feature-basierte Prüfmodellierung, in: VDI-Berichte Nr. 1569, 2000, 199–212. VDI-Verlag, Düsseldorf.

[11] Matthes, J.; Bullinger, H.-J.; Warschat, J.: Featureorientierte Produktbeschreibung zur konstruktionsbegleitenden Montageplanung, VDI Berichte Nr. 1322, 135–160, VDI Verlag, Düsseldorf, 1997

[12] Molloy, E.; Yang, H.; Browne, J.: Feature-Based Modelling in Design for Assembly, in: International Journal of Computer Integrated Manufacturing 6 1/2 (Special Issue Features, Hrsg. K. Case, N. Gindy), 1993

[13] Shah, J. J., Mäntylä, M.: Parametric and Feature-based CAD/CAM, John Wiley & Sons, Inc., 1995. – ISBN: 0-471-00214-3

[14] Siemens PLM Software, JT File Format Reference Version 8.1 Rev-B. http://www.jtopen.com/docs. accessed 2009

[15] Siemens PLM Software, White paper - open product lifecycle data sharing using xml. accessed 2009. http://www.plm.automation.siemens.com, 2007.

[16] Tassi, E.: Knowlegde-Features für die Produkt- und Technologieentwicklung in umformenden Prozessen, Dissertation Universität Chemnitz, 2004.

[17] Vanja, S.; Weber, Chr.; Bley, H.; Zeman, K.: CAx für Ingenieure; Eine praxisbezogene Einführung, 2.Auflage, Berlin Heidelberg, Springer-Verlag, 2009.

[18] VDI: VDI 2218 – Information technology in product development: Feature Technology, March 2003

[19] Pahl, G.; Felhusen, J. et al: Konstruktionslehre, Grundlagen erfolgreicher Produktentwicklung, 2. Auflage, Springer-Verlag, Berlin Heidelberg, 2005.