INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

DESIGN PROCESS ACCELERATION BY KNOWLEDGE-BASED ENGINEERING IN AUTOMOTIVE AND AEROSPACE INDUSTRY

J. Stjepandić, S. Rulhoff, W. J. C. Verhagen, H. Liese and P. Bermell-Garcia

Keywords: new product development, virtual product creation, artificial intelligence, KBE, knowledge-based engineering, knowledge management, template

1. Introduction

During the development, production and operation of products, knowledge is created in various forms. The central role of knowledge in all aspects of the product lifecycle is well recognized, as is its function as a driver for the competitiveness of an organization through the capability for effective action [Birkinshaw and Sheehan 2002]. Knowledge systems capitalize on various research streams to capture, structure and (re-)use product and process knowledge within an organisation.
One of the most promising research and application fields in this context is Knowledge-Based Engineering (KBE). KBE applications aim to reduce time and cost of new product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and re-using product and process knowledge [Liese 2004], [La Rocca 2012]. Dedicated KBE systems are available for development of KBE applications, but vendors of Computer-Aided Design (CAD) systems are offering an alternative in the guise of so-styled knowledge or KBE template solutions. Though the two approaches have some fundamental differences [La Rocca 2012], the following questions arise from an industrial perspective: the two approaches have some fundamental differences. Whereas the academic foundation for KBE has been established to some degree and is being improved upon, the use of templates in CAD systems has received relatively little attention in literature and is often grouped in with KBE literature. As such, the following exploratory questions arise: Which types of templates can be distinguished? Which challenges are encountered in the development of templates? What is the comparative performance achieved by knowledge templates with respect to 'true' KBE applications?
The presented research explores these issues and highlights recent work from an industrial perspective. Two industrial cases are introduced in which template functionality is implemented within a CAD system. The developed template functionality addresses part of the challenges identified in Section 2. The findings of the industrial cases are subsequently compared in a dedicated discussion section. Finally, ongoing and future research work with respect to knowledge templates in CAD systems is discussed.

2. Theoretical context

KBE can be seen as an extension of Knowledge-Based Systems (KBS) into the design engineering domain, where KBE adds facilities for geometry manipulation and data handling to the acquisition mechanisms, structured knowledge base(s) and reasoning mechanisms incorporated in KBS [La Rocca

2012]. Though KBE has its roots in the 1980's and has been used throughout the years in the automotive and aerospace industries in particular, developments such as the commoditization of computing and methodological advancements in KBE [Stokes 2001] have sparked a renewed interest from both academia and industry, including small-to-medium enterprises (SME's) [Lovett 2000]. Part of this interest is prompted by the tangible benefits achieved by KBE applications, as reported in literature [Verhagen et al. 2012] – time savings for complex design tasks can be as high as 99%.

Commercial offerings for the development and implementation of KBE solutions are typically categorized as being 'true' KBE systems or 'augmented CAD (Computer-Aided Design)' systems. True KBE systems are characterised by the native use of dedicated programming languages and focus on rule and knowledge capture; for an extended discussion, see La Rocca [2012]. In an effort to match the benefits offered by 'true' KBE systems, vendors of Computer-Aided Design (CAD) systems have made a conscious effort over the past decade to integrate KBE-like functionality into their geometry-focused solutions. This is done through the use of knowledge-based engineering templates. The resulting 'augmented CAD' systems offer similar capabilities in terms of design automation, while being positioned as user-friendly alternatives to 'true' KBE systems, which require programming expertise. On the downside, the augmented CAD systems are either slower than true KBE applications, or require significant programming effort to realize similar fast automation results [La Rocca 2012].

Despite the differences between KBE systems and templates in CAD systems, both approaches to design automation face a number of common challenges. These include black-box implementation and (non-)transparency of knowledge bases, maintainability of knowledge and applications, ad-hoc development, and a notable lack of (quantified) justification [Verhagen et al. 2012]. However, in the context of this industrial paper the focus lies on the characterisation and application of templates in CAD systems, for product validation and process representation in particular.

3. Implementation of knowledge templates

In industrial practice, various templates can be meaningfully applied to support product design. Templates can be characterized using product and/or process breakdown approaches. As an example, characterization may include part templates, assembly templates, validation templates, process chain templates and system templates. Part and assembly templates, often complemented by design rules, are geared to the parts and sub-assembly generation. Often, they are used in connection with skeleton model methods which describe references and parameters that are necessary to shape and instantiate parts in a sub-assembly. The validation template serves for the testing of complex rules (e.g. legitimate duties). In contrast, a process chain template connects different domains (e.g. design and manufacturing). In addition, system templates connect descriptions of functions and behavior with the shape. Different template types can be combined with each other. In the remainder of this Section, two use cases from industry pertaining to a validation template and a process chain template are presented.

3.1 Use case 1: Validation template for application in automotive industry

Validation templates are a very promising application of KBE because the workflow is defined by law. Since the license for a passenger car is subject to many international rules, norms and standards, the KBE template CAVA (CATIA V5 Automotive Extensions Vehicle Architecture) was developed by Audi, BMW, Daimler, Porsche and Volkswagen to ensure car design and legal compliance during the entire design process - from the concept phase to the homologation [Rohwäder 2007].

As an additional CATIA V5 workbench CAVA creates reference or support geometry representing design space, clearance areas, or fields of vision required to support draft and design. During the concept phase CAVA provides the boundaries for several design aspects, performs automatic checks for legal conformity and reports deviations. CAVA includes a complete set of validation procedures like rear view mirror, viewing fields, security belts, underfloor clearances, lamp positions, pedestrian protection, and much more. Finally it validates that standards have been followed and creates reports to be used for homologation of a car.

Nevertheless there is a strong need for further automation of the validation procedure in the concept development of passenger cars, in particular in case of modular architecture and multi-brand product strategy. Hence, validation is one of the tasks that fully harness the benefits of the modular product strategy. Otherwise, it shall preserve the quality of design solutions. The concept development of a passenger car runs by creating a 3D master model and using many knowledge based assistants that support the designers in dedicated design tasks: silhouette of a car, design of wheel housing, pedestrian protection, gap tolerance area, ejection mitigation, barriers for bumper tests protection, seat belt numerical simulation [Brüning and Liese 2013]. With this approach the following strategic advantages are addressed:

- Time savings: CAD model can be built and changed *faster*.
- Process quality: Earlier quality assurance with sound geometry data.
- Standardization: Design specifications and guidelines are used consistently.
- Efficiency: CAD models can be built and changed with less effort.
- Knowledge management: Knowledge-based assistants integrate processes, design and methods.
- Modularization: Overall task is broken down to manageable sub-modules with well-defined interfaces.
- Usability: Intuitive GUI with help feature and quick reference guides.

There is strong demand for automatic cross-section generation in 3D models. Creation and evaluation of conceptual designs takes place in 3-D CAD. Pre-defined 2-D design cross sections of traditional package plans still form an indispensable part of the development process. Whenever new CAD software is introduced, the methods of interactive 2-D section derivation with special applications are preserved. Due to growing model sizes, design engineers have had to accept constantly increasing amounts of manual effort and waiting times for this purpose. The assurance of sustainable faultlessness – i.e. currency, accuracy, repeatability – for regular cross-section generation is becoming increasingly difficult with the growing number of projects.

To fulfill increasing process requirements, cross sections of all conceptual DMU data shall be generated fully automatically during the night. The automatic cross-section generator shall provide project coordinators with extensive configuration options for classifying the cross-section geometries for various graphic representations. Various benefits are expected: substantial time-saving for concept developers, obsolesce of manual cross-section maintenance and a significant increase in process quality.

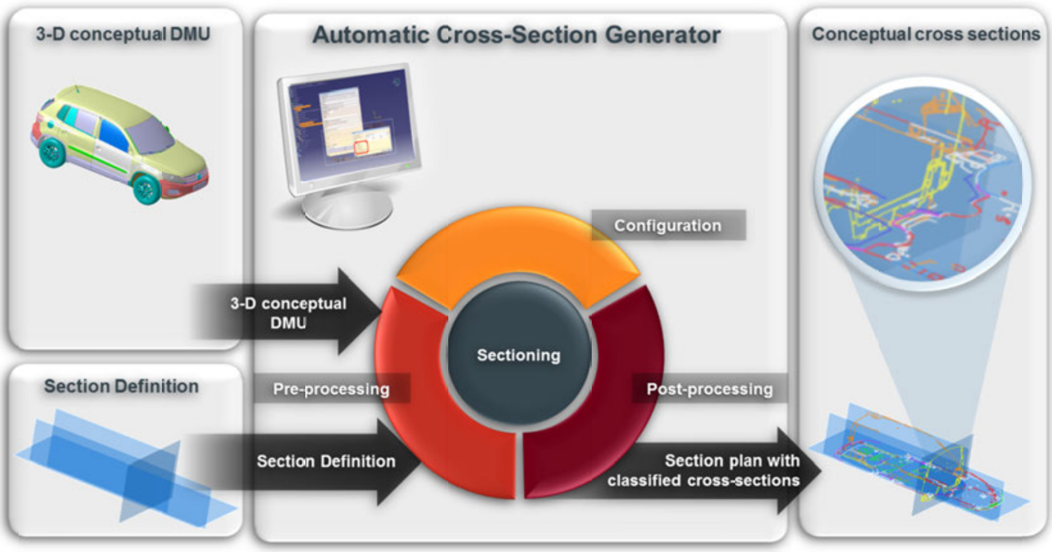

Figure 1. Architecture of Automatic Cross-Section Generator

To support this, a software tool called Automatic Cross-Section Generator has been developed, offering various configuration options for automatic processing of car concepts. It is implemented as an additional workbench with CAA V5 application programming interface for CATIA V5 and fully embedded in the CAD infrastructure. It consists of four main building blocks: configuration, pre-processing, sectioning and post-processing.

Configuration is maintained via graphical user interface. It facilitates the setup of global options, section-job definition, continuous sections, special sections and section of reference cars (from previous or other current product families). Additionally it ensures typical administration tasks: classification/configuration of sectioning results, editing job definitions and defining sets of job definitions.

The building block pre-processing prepares the operational sectioning carrying out the following tasks: analysis of 3D product and section definitions (valid product-structure, duplicate detection, check of metadata, valid input for sectioning process, space analysis) and modification recognition while minimized calculation time is achieved based on detection of modified geometry.

The main building block of the automatic cross-section generator is sectioning that processes the input CATIA package following the complex design rules. The input 3-D conceptual DMU CATProduct remains unmodified, while cross-sections are created in the CATProduct containing the section plans. Report files are generated and put in pre-defined directory.

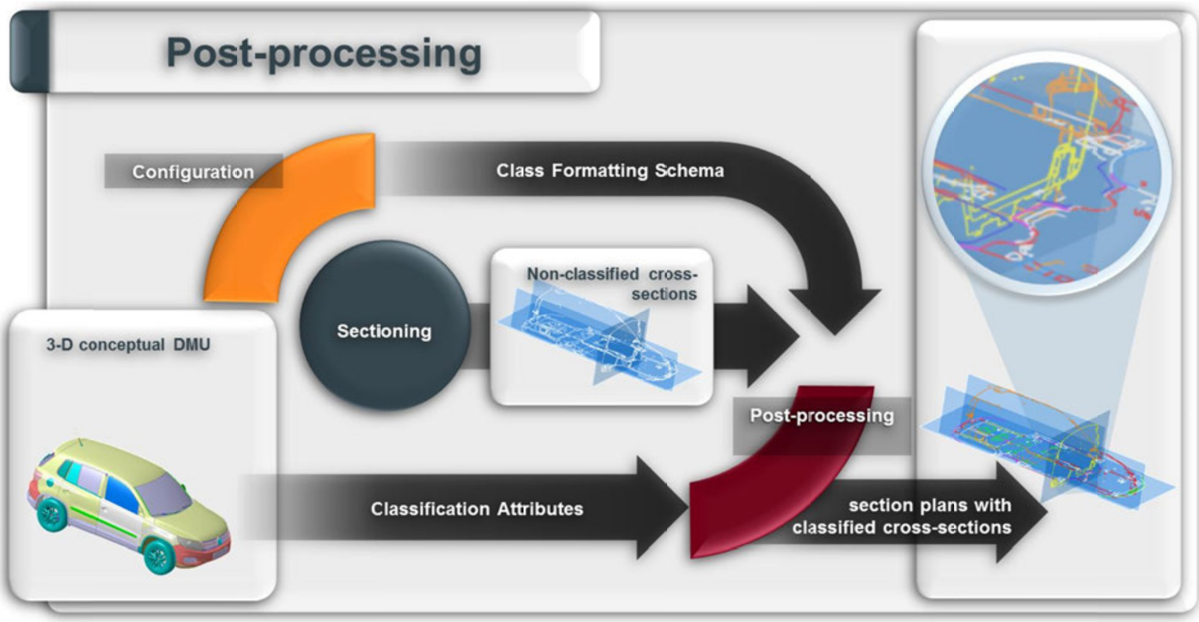

Figure 2. Postprocessing of Cross-Section Results

After the needed sections are generated, the resulting cross-section data is stored in the 3-D CAD structure during post-processing (Figure 2). Modification of resulting cross section geometry is based on classification rules: structure, geometrical sets, properties, line type, line thickness and line color.

To leverage the operation and monitoring, strong optimization of loading and memory strategies is implemented. Batch operation is optimized by subdivision into different processes what ensures high performance as well as decrease of maximal memory usage. Realization of different report and protocol levels round up the user interaction.

Introduction of the automatic cross-section generator and batch operation on a high-performance server signifies a substantial time-saving for the concept developers. Thus, significant process quality improvement by automation and permeation of the whole vehicle project at predefined levels without user interaction are achieved. Resulting time and capacity savings are significant, while the move from manual to fully automated work has become possible.

3.2 Use case 2: Process chain template for application in aerospace industry

In the aerospace domain, the introduction of life cycle constraints into the design phase is a major theme, as encompassed by the Design for X philosophy. A major representation of this is Design for Manufacturing. Bermell-Garcia et al. [2012] have developed a knowledge-based application to

optimize a composite wing cover conceptual design for ply continuity through the blending of ply stacking sequences. This design task incorporates design and manufacturing constraints that are applied to the wing panel structure in order to optimize it for weight, cost and manufacturability. In contrast to the previously published work, this use case focuses on the (automated) process workflow, where the implemented solution can be considered as a process chain template.
The existing process for solving the ply continuity problem at hand is contained within a solution known as mPDA (manufacturable Ply Design and Analysis). mPDA is a solution developed in-house at the industrial partner. It has been implemented using Microsoft Excel and VBA and interfaces with CATIA V5. The associated top-level design task (optimize wing cover for ply continuity) requires input from the structures department in the form of a ply specification file that specifies the wing cover grid, grid cell thicknesses and associated stacking sequences (a so-called fishtail plot). This input is processed by the ply optimizer tool (mPDA) to generate an optimized ply specification. The ply optimizer takes into account various design and manufacturing constraints.
The top-level task can be split up into four subtasks – see Figure 3. The preparation task (A1) takes panel sizing information (minimum required thickness and thickness law per grid cell) from the ply specification file to generate a catalogue of stacking sequences. This catalogue is used in the processing task (A2), which uses the ply specification file as an input for the generation of ply fishtail plots – a set of fishtail plots is generated to indicate grid coverage per ply, instead of one specification file containing thickness and/or stacking sequence per grid cell. In effect, the specification file is decomposed into a set of (virtual) fishtail plots indicating the grid coverage per ply layer. The optimization task (A3) applies the design and manufacturing requirements and constraints to the ply fishtail plots to configure optimized ply fishtail plots. Finally, the post-processing task (A4) uses these plots to put together an optimized ply specification file, where the wing cover has been optimized for ply continuity.

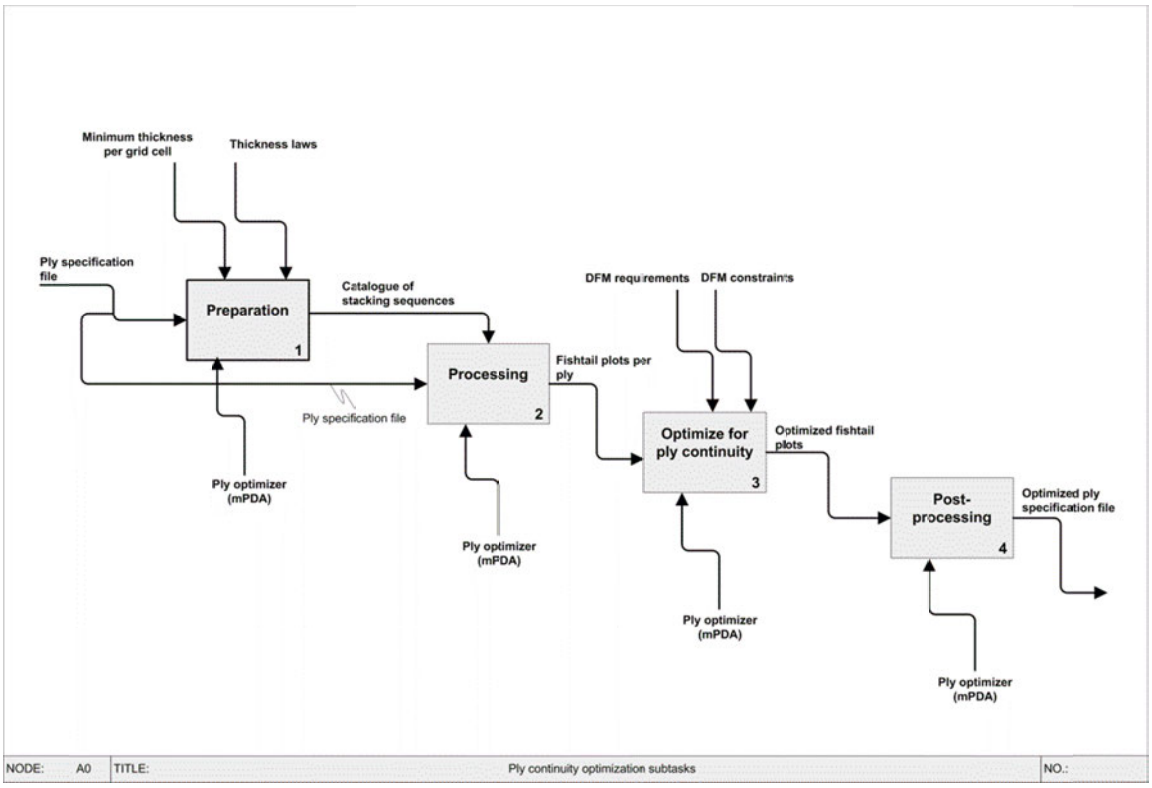

Figure 3. IDEF0 A0 diagram for ply continuity optimization subtasks

The process chain as represented in Figure 3 is formalized as part of a dedicated ontology, shown in Figure 4. It consists of two central perspectives:

- Enterprise Knowledge Resource (EKR): a task-oriented container of the knowledge, subtasks and output associated with an engineering task. In other words, an EKR can be used to represent the content of a task.
- Product-Process-Resource: classes to represent the product(s), process(es) and resource(s) an EKR is associated with. In other words, these classes represent the context of a task.

As the task is separated from the domain knowledge, it is possible to independently update either element. Given the association of the task with its context, it becomes easier to retrieve and access a task as well as its individual elements. In short, maintainability and usability of a knowledge-based application are improved.

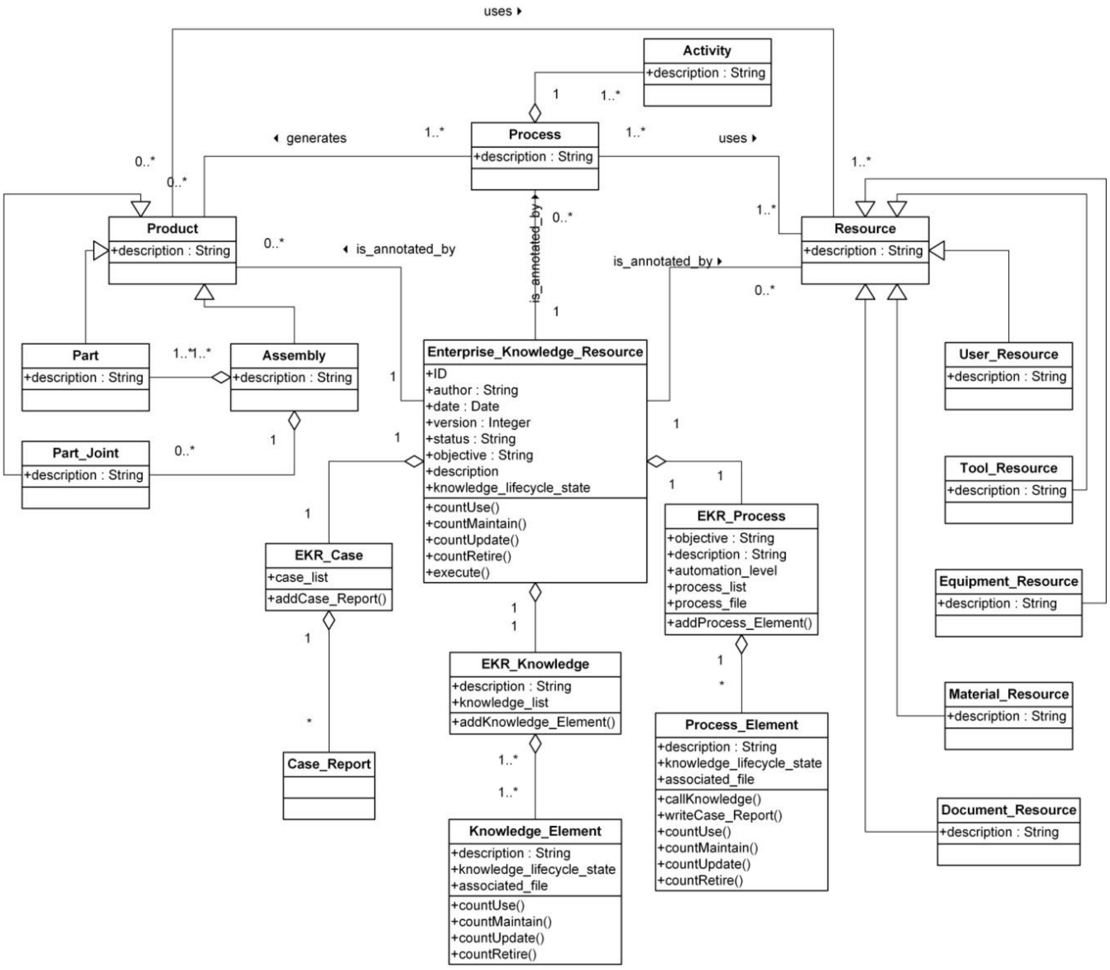

Figure 4. Knowledge Lifecycle Ontology [Verhagen 2013]

The ontology has subsequently been used as the semantic backbone for development of a knowledge-based application. Using the preceding considerations, an EKR class diagram has been modelled for this specific case study and associated task. The UML class diagram is shown in Figure 5.

The knowledge-based application optimizes for ply continuity using an automated solution, while retaining the required product and process knowledge. This knowledge includes over 30 design and manufacturing constraints. Furthermore, a genetic algorithm has been implemented in FORTRAN to perform the optimization of ply continuity. To implement the solution, two main architectural elements have been devised:

- **EKR Environment for Learning by Doing (eLBD):** The environment for learning by doing (eLBD) is a web solution aimed at supporting end users. eLBD is based on a knowledge management tool called Ardans Knowledge Maker (AKM). Specific models for the representation of knowledge and process elements have been implemented within AKM to enable the construction of EKRs which package the process and knowledge elements and the cases. For this use case, AKM contains an EKR which holds the design and manufacturing knowledge related to the ply optimization problem, as well as a process representation that calls the process workflow when executing the EKR (see below). The resulting process template is consequently not implemented in CATIA, but interfaces with it through the ply specification file.

- **Executable environment for Learning by Doing (xLBD):** The executable environment for learning by doing (xLBD) is a solution to enable the remote execution of EKRs through a web service approach. xLBD uses several software applications and languages (Apache Tomcat web server, Java, AKM web services and Phoenix Integration Model Center®) to deploy EKRs as web services. Users can access and execute the software remotely, so they do not require a dedicated installation of software on their desktops. In particular, the process workflow is modeled and automated using Model Center, where the tasks highlighted above are implemented in a sequential order. This workflow is initiated through the eLBD environment.

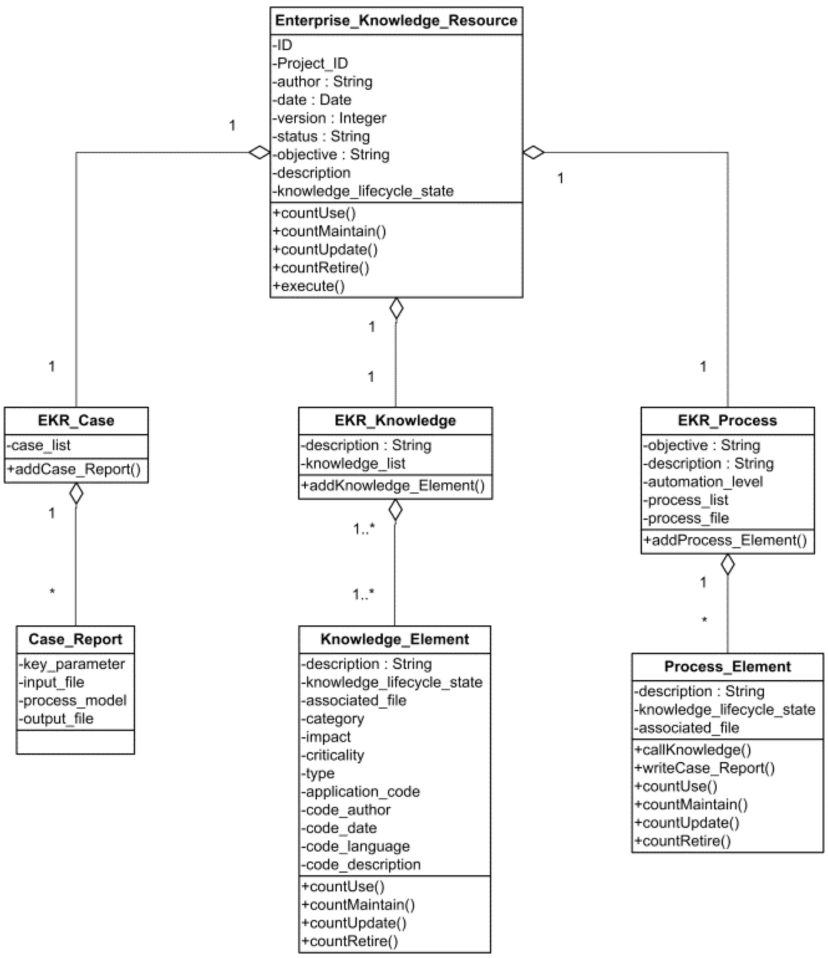

Figure 5. EKR class diagram (UML) for design case study

The implemented knowledge-based solution is able to deliver composite wing conceptual designs optimized for manufacturing. Runtime of a single solution is around 1 minute, as opposed to one to several hours (incorporating significant variance dependent on the provided specification file and manual interventions required within the CAD system) for the previous, highly manual solution. Evaluation and trade-off of the conceptual designs is supported by the construction of Pareto fronts using weight, cost and manufacturability objective parameters. The solution is fully automated and supports independent update of the task knowledge and process workflow. As such, maintainability of the knowledge-based application is taken into account.

4. Discussion

KBE technology has become widely accepted in many industries. Especially for design processes with a high percentage of repetition KBE may demonstrate significant benefits, and justifies the name design automation. In general, the benefit of KBE is extra high if products feature a high variance of a basis development, as much as possible single steps of the engineering process can be described by algorithms, and operations are cross-divisional (e.g. product design or tool design). By using a KBE application or template routine tasks are transformed into an automated process with a minimum of user assistance. The high one-time effort for the development of a template loses importance the more the template is used. All these criteria are fulfilled in the presented use cases.

KBE is still subject to intensive research and practical improvements. The valuation of KBE can be facilitated in two directions: KBE as autonomous procedure (application) and KBE as component of the business process as a whole. This first direction is demonstrated in the automotive case study. The second direction is partially included in the aerospace case study.

When considering KBE autonomously, a number of developments have put KBE on a stronger footing over the last decade. In particular, the development and uptake of KBE-dedicated languages, the uptake of KBE language or template functionality in most major CAD systems, and the development of supporting methodologies such as MOKA [Stokes 2001] have strongly contributed to the uptake of KBE. If a KBE application is developed close to a dedicated CAD system, as shown in our automotive case study, its flexibility to be used with other CAD system is heavily inhibited. However, a number of challenges remain to be solved.

First of all, the justification of KBE application development, use and maintenance is currently not supported by scientific approaches [Verhagen et al. 2012], though recently advances have been made to support the identification of knowledge change characteristics [Verhagen 2013] and to offer methodological support [Johansson et al. 2013].

Second, current KBE applications or templates (within CAD systems) only weakly address the issue of workflow integration. In our automotive use case it is done by a dedicated software application which must be also updated in case of update of the KBE application. The European iPROD project [iProd 2013] aims to address this through the development of models and tools for workflow integration into KBE. The project iPROD also considers KBE as a service (Software-as-a-Service, SaaS), thereby addressing a third challenge in KBE research, which is to 'support web collaborative solutions and open source initiatives' [La Rocca 2012].

A final research challenge is to address the 'black-box' phenomenon of KBE development and use [Verhagen et al. 2012]: supporting the integration of code and documentation generation [Liese 2004] as well as explicitly linking knowledge base structure and meaningful content with the KBE application elements and code.

When considering KBE as a singular component of a business process, many issues have arisen in the past and still remain unresolved. As KBE uses either additional CAD entities or additional software, which must be independently installed, in the case of a PDM-controlled development process, KBE systems must be subjected to the PDM system, as practised in our automotive case study. That is a serious issue because commercial PDM systems do not sufficiently support the KBE workflow [Katzenbach et al. 2007]. Therefore, a certain self-limitation of the used KBE functionality is necessary to remain in conformance with PDM, at least until PDM systems are able to support the whole KBE functionality.

Another issue is the complexity of the KBE data model and the corresponding relationships between singular entities. Although KBE can be applied easily like a blackbox, misunderstandings and mistakes can arise if the user has to derive the CAD model created by instantiation of a KBE template using complex relationships he is not familiar with. For example, more than 2,500 links are needed to define an entire body-in-white structure within a concept template [Katzenbach et al. 2007]. This link management gives the capability of dividing complex structures into template- based and usable part structures. This issue could be resolved by use of augmented reality (AR) techniques that enable the multimedia representation of knowledge, especially interactive animation of 3D CAD models [Januszka et al. 2012]. AR provides the changing views of virtual data – especially 3D models – in a real environment and allows the user to better understand the presented virtual data and knowledge in a more comprehensive way. Therefore, visualization of relationships in engineering remains a subject of research.

5. Conclusions and Outlook

As the preceding sections and use cases indicate, the use of KBE applications and templates can achieve significant results through automation of complex engineering tasks, significantly decreasing engineering task time while retaining product and process knowledge and increasing the robustness and process stability. Recent developments like CATIA V6™ allow use of templates for each module in this package allowing automation of (theoretically) each engineering task. It could be used as one of basic technologies for systems engineering too [Fukuda et al. 2013]. In such way design automation becomes more ubiquitous.

A breaking point preventing the comprehensive use of KBE lies in engineering collaboration. Commercial KBE applications are not prepared for the exchange of templates on a regular basis (e.g. within the supply chain). Our both use cases underlie the same constraint. It helps, although not intentionally, to protect the intellectual property stored in this templates [Liese et al. 2012]. Therefore, each partner in the supply chain is forced to act autonomously. Furthermore, if KBE entities shall be exchanged between two applications (e.g. different CAD systems), it would fail because there is no standard interface for such an exchange. Apart of the first draft of the KBE Services for PLM done by the Object Management Group (OMG) [Bermell-Garcia 2007], there is no known standardization activity.

Finally, when considering the introduction and use of KBE applications and templates, the adaptability and maintainability of developed applications and templates remains a point of concern.

References

Bermell-Garcia, P., "A Metamodel to Annotate Knowledge based Engineering Codes as Enterprise Knowledge Resources", PhD dissertation, Cranfield University, 2007.

Bermell-Garcia, P., Verhagen, W. J. C., Astwood, S., Krishnamurthy, K., Johnson, J. L., Ruiz, D., Scott, G., Curran, R., "A Framework for Management of Knowledge-based Engineering Applications as Software Services: Enabling Personalization and Codification", Advanced Engineering Informatics, Vol. 26, 2012, pp. 219-230.

Birkinshaw, J., Sheehan, T., "Managing the knowledge life cycle", MIT Sloan Management Review 44, 2002, pp. 75-83.

Brüning, H. C., Liese, H., "Prozesssichere Methoden für die Virtuelle Fahrzeugkonstruktion in der Pkw-Konzeptentwicklung der Volkswagen AG", ProSTEP iVip Symposium, Hannover, April 16 - 17, 2013.

Fukuda, S., Lulić, Z., Stjepandić, J., "FDMU –Functional Spatial Experience Beyond DMU?", In: C. Bil et al. (eds.), Proceedings of 20th ISPE International Conference on Concurrent Engineering, IOS Press, Amsterdam, 2013, 431-440.

iProd, "Improving the Product Development Process (PDP - iPROD project)", Retrieved 20-03-2013, from <http://www.iprod-project.eu/index>.

Januszka, M., Moczulski, W., "Acquisition and Knowledge Representation in the Product Development Process with the Use of Augmented Reality", Proceedings of the 19th ISPE International Conference on Concurrent Engineering - Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment, J. Stjepandić, G. Rock, and C. Bil (Eds.), Springer, London, 2013, pp. 219-229.

Johansson, J., Elgh, F., "How to Successfully Implement Automated Engineering Design Systems: Reviewing Four Case Studies", In: Bil C et al. (eds.) Proceedings of 20th ISPE International Conference on Concurrent Engineering, IOS Press, Amsterdam, 2013, pp 173-182.

Katzenbach, A., Bergholz, W., Rohlinger, A., "Knowledge-Based Design – An Integrated Approach", In: Krause F.L. (ed.) The Future of Product Development. Proceedings of the 17th CIRP Design Conference, Springer-Verlag, Berlin Heidelberg, 2007, pp. 13-22.

La Rocca, G., "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design", Advanced Engineering Informatics, Vol. 26, 2012, pp. 159-179.

Liese, H., „Wissensbasierte 3D-CAD Repräsentation", Shaker Verlag, Aachen, TU Darmstadt PhD thesis, 2004.

Liese, H., Rulhoff S., Stjepandić J.,"Enhancing Product Innovation by Implementing Intellectual Property Protection into the Virtual Product Creation", In: Concurrent Engineering Approaches for Sustainable Product

Development in a Multi-Disciplinary Environment, Proceedings of the 19th ISPE International Conference on Concurrent Engineering, Stjepandić J, Rock G, Bil C (eds.), Springer-Verlag, London, 2013, pp. 267-278.
Liese, H., Stjepandić, J., "Konstruktionsmethodik: Wissensbasierende 3D-CAD-Modellierung", CAD/CAM Report, Dressler Verlag, Heidelberg, 10, 2004 retrieved 15-05-2013, from <http://www.prostep.com/de/prostep/medien/cad-cam_kbe.htm>.
Lovett, P. J., Ingram, A., Bancroft, C. N., "Knowledge-based engineering for SMEs -- a methodology", Journal of Materials Processing Technology, 107(1-3), 2000, pp. 384-389.
Rohwäder, T., "CAVA Erleichtert die Einhaltung der Gesetzlichen Vorschriften", CAD-CAM Report, 2007, pp. 20-26.
Stokes, M., "Managing Engineering Knowledge – MOKA: Methodology for Knowledge Based Engineering Applications", Professional Engineering Publishing Limited, London, 2001.
Verhagen, W. J. C., "An Ontology-Based Approach to Knowledge Lifecycle Management within Aircraft Lifecycle Phases", PhD dissertation, Delft University of Technology, 2013.
Verhagen, W. J. C., Bermell-Garcia, P., Van Dijk, R. E. C., Curran, R., "A critical review of Knowledge-Based Engineering: An identification of research challenges", Advanced Engineering Informatics, Vol. 26, No. 1, 2012, pp. 5-15.

Dr. Josip Stjepandić, Head of Business Unit 3D Product Creation
PROSTEP AG
Dolivostrasse 11, 64293 Darmstadt, Germany
Telephone: +49 6151 92 87 363
Telefax: +49 6151 92 87 326
Email: josip.stjepandic@opendesc.com
URL: http://www.prostep.com