KBE-PLM Integration Schema
for Engineering Knowledge Re-use and Design Automation

Jullius Cho[1(✉)], Thomas Vosgien[2], Thorsten Prante[2],
and Detlef Gerhard[1]

[1] TU Wien, Vienna, Austria
{jullius.cho,detlef.gerhard}@tuwien.ac.at
[2] V-Research GmbH, Dornbirn, Austria
{thomas.vosgien,thorsten.prante}@v-research.at

Abstract. The re-use of product knowledge is vital to the development of Knowledge-Based Engineering (KBE) systems and to the deployment of Product Lifecycle Management (PLM) strategies. This paper addresses the challenges related to KBE-PLM systems integration in order to unlock engineering knowledge from proprietary representations and to manage the lifecycle of KBE models as well as their usage by different design automation applications. Essential constituents of product knowledge are identified and analyzed and the concepts of design intent and design rationale are re-introduced as key enablers to re-use this product knowledge in the appropriate KBE context. The paper introduces a KBE-PLM integration framework including a platform-independent Open KBE repository structured according to the KBE-PLM integration schema. This schema is a multi-layer neutral product and knowledge data model designed for integrating information from parameterized Computer-Aided Design (CAD) models, rule-based KBE systems and PLM systems.

Keywords: Knowledge-based engineering · Design automation · Knowledge re-use · Design intent · Design rationale

1 Introduction

As part of the implementation of PLM strategies, contemporary computer-based tools should provide engineers with means to better handle the processes involved across the product lifecycle. Within this context, there is considerable pressure on engineers to improve their product development processes in order to deliver complex and high-quality products and services to the market rapidly and at low cost [1]. One approach to achieving this improvement is engineering design automation, where interdependent engineering design processes are subject to partial or full automation. If the focus of the automation is on repetitive tasks, this is called Knowledge-Based Engineering (KBE) [2]. For its implementation, product and process engineering knowledge needs to be identified, acquired, formalized, processed and re-used. However, on the one hand, knowledge acquisition is still a bottleneck and, on the other hand, the formalized engineering knowledge is too often encapsulated in CAD models

and inference engines of KBE systems, which are not easily accessible outside vendor-specific application environments. Thus, re-use of formalized knowledge across KBE applications, and even within the same KBE application across time, is an issue. This is reflected in the following shortcomings of KBE as identified in [2]: First, case-based, ad hoc development of KBE applications, second, a tendency toward development of 'black-box' applications (i.e. formalized knowledge is not explained or contextualized, so not related to design intent – with some exceptions such as [3]) and, third, generally, a lack of knowledge re-use is observed – where these three reported shortcomings are interrelated. Compared with this, PLM support entails the modeling, capturing, manipulating, exchanging and using of information all along product life cycle decision making processes, across all application domains [4]. Therefore, a strong focus of a PLM strategy should be on knowledge re-use and it doesn't come as a surprise that integration of KBE and PLM systems has been identified to provide advantages [5, 6] for product development. For instance, the following deficiency could therewith be overcome: Although many CAD systems provide basic design automation capabilities which allow parts, assemblies, and drawings to be parametrically generated and changed, the capabilities for consistent changes to product configurations generally remain restricted because knowledge in repositories of KBE systems is not linked to the product information/definition managed in PLM systems [5]. Still, there are major challenges for overall KBE-PLM integration and its interoperability enabler: one, the "ability to unlock engineering knowledge from proprietary representations" and, two, the ability to manage the lifecycle of KBE models" [6]. Overcoming these challenges will result in improved support for the development and maintenance of KBE systems for design automation and hence for the overall engineering design process efficiency. In this paper, we propose a KBE-PLM integration framework enabling to retrieve and integrate the information and knowledge from CAD, KBE and PLM systems in order to store and manage them in a platform independent Open KBE repository structured according to the KBE-PLM integration schema. The goal is to be able to re-use this knowledge for different KBE applications for design automation. Section 2 discusses relevant background and state of the art. The architecture of the framework and the KBE-PLM integration schema are presented in Sect. 4 and discussed in Sect. 5, before concluding the paper and presenting lines of possible future work.

2   Background

2.1   Product Knowledge, Design Intent and Design Rationale

There are several methodologies which have been developed and/or used for the design, development and deployment of KBE systems, such as MOKA [7], Com-monKADS [8], and KNOMAD [9]. While most of them have addressed the issue of knowledge capture and structuring/formalization, none of them has yet integrated the concepts of design intent and design rationale to facilitate better re-use. According to [10], product knowledge includes function, structure and behavior models of an artifact along with product specifications, requirements, and constraints and includes various kinds of relationships between parts and assemblies [11]. This definition is in line with

the Function-Behavior-Structure (FBS) framework proposed by [12]. In [11], the authors also include design process knowledge as part of the product knowledge, which can be encoded as methods in a product representation and provides mechanisms for realizing design details at various stages of the product lifecycle and in [10] process knowledge is defined as design rules, strategies, automated model mappings, analysis scripts and problem-solving algorithms. In making the connection between product and design knowledge on the one hand and design intent and design rationale on the other, it must be noted that there is some ambiguity regarding the definitions of the two later terms [11]. While design intent is captured in a set of functional and non-functional requirements, the design rationale encapsulates both the semantics and conditions for all the mappings and transformations from requirements to architecture (i.e. structure) [14]. Thus, design rationale relates to and describes the process of designing ('design logic') and is related to the design intent which represents the reason(s) that justify design decisions. In this paper, design intent and rationale will be defined as follows (see Sect. 3.2): design intent is a concept that, based on a specific usage context, enables one to know which product structural representation and related design rationale to use to fulfill a specific functional intent. Design rationale refers to the design rules, automated model mappings, analysis scripts and problem-solving algorithms required to fulfill a specific intent.

2.2 Model-Based Definition (MBD) and KBE-PLM Systems Integration

The central concept embodied in MBD is that the 3D product model is the most appropriate vehicle for delivering all of the detailed product information necessary for downstream organizations to perform their portion of the product creation [13]: CAD models are enriched with explicit and implicit knowledge which needs to be extracted, formalized and managed for re-use in different contexts. However, extracting knowledge encapsulated in CAD models remains a challenge. Gerhard and Lutz [14] propose a knowledge acquisition and knowledge formalization assistant (KAFA) which employs the design structure matrix technique to capture product knowledge directly from CAD models and enable experts to formalize the design logic. Using CAD models as the main container of product knowledge stresses the necessity to be able to extract and re-use this knowledge in heterogeneous environments such as KBE and PLM systems. However, current MBD approaches do not address the development of neutral product data representations which enable links between representations of product definition, design intent and design rationale. Combining both MBD and KBE-PLM approaches is required to efficiently encapsulate design knowledge in CAD models and at the same time to extract the knowledge for re-use in faster platform-independent KBE systems development or in the appropriate design context and related activities.

2.3 KBE-PLM Systems Interoperability

An example of a standardized modeling framework is the Model-Driven Architecture (MDA) for portability, interoperability and reusability between IT systems [15].

In order to achieve integration and interoperability, transparent access to proprietary data and services is needed. Moreover, a neutral representation of product information will help to increase interoperability. Truyen [16] addresses these concerns and specifies three default viewpoints on a system: computational independence, platform independence and platform specific models. Correspondingly, the framework proposed in this paper (Sect. 3), needs to be CAD, PLM and KBE systems independent. Thus, we have considered the following:

- Neutral product data representations to ensure data interoperability between CAD and PLM systems.
- Neutral knowledge representations to ensure the interoperability between KBE systems.
- A CAD abstraction layer enabling managing multi-CAD APIs giving access to the various CAD systems capabilities and enable inference engines to be CAD independent.

There are works dedicated to the exchange of data between CAD and PDM systems as well as the exchange of CAD and configuration data information between KBE and PLM systems such as [17]. The Standard for the Exchange of Product Model Data (STEP)[1] covers many aspects of product data exchange in the context of product development. The emerging STEP Application Protocol (AP) AP242 [18] merges AP203 and AP214 which are currently the most implemented for CAD data exchange between existing commercial CAD systems. The PDM Schema is a core set of entities in STEP that support the mapping of concepts for Product Data Management (PDM) systems interoperability integrating different STEP generic and application resources among which Parts 41 (fundamentals of product description and support), 42 (geometric and topological representation), and 44 (product structure configuration). The transfer of associated geometric interrelationships, parameters, and constraints, which is of particular importance to our proposal and which is not yet implemented in any application, is addressed by the Integrated-Application Resource 108 [19] and also in [20]. In [21], the author proposes a multi-layer design analysis systems integration framework integrating various concepts defined in the aforementioned STEP AP and parts as well as concepts from other neutral product data representations such as the CPM [22]. However, these endeavors have not solved the problem of being able to explicitly and unambiguously share geometric parameters and the design intent/design rational for knowledge re-use.

Concerning neutral KBE knowledge representations, some knowledge sharing initiatives in the beginning of the 90's have led to the implementation of some neutral knowledge formats in specific KBE applications such as the Knowledge Query and Manipulation Language (KQML) [23] and the Knowledge Interchange Format (KIF) [24]. However the adoption and implementation of these formats are restricted to few applications and their development and maintenance seem to have been stopped.

---

[1] NB: To respect the length requirement of the paper, some of the mentioned parts of the ISO STEP standard are not in the reference list.

More recently, the Rule Interchange Format (RIF) [25] has been developed to enable knowledge representation and information processing on the Web.

Very few works can be found in the literature regarding the development of platform-independent KBE systems. However we can quote the recent work from [26] in which authors propose a software framework using ontologies for representing design knowledge and for implementing platform-independent KBE systems within the aerospace industry. An essential element of the proposed framework and which will also be required for our framework is the use of APIs and GUIs that enables direct integration and communication between a KBE knowledge base, different CAD systems, GUI application and end-user.

Finally, in 2006, the Object Management Group (OMG) submitted a Request for Proposal for an international standard for KBE Services for PLM [27]. This standard intended to facilitate the integration of KBE applications in a PLM environment. To date, no answer to this RFP and no OMG specification for KBE services exist. This paper do not pretend to answer the request for proposal, but the framework and the schema presented in the next section could serve as a basis for future work in that direction.

3 The KBE-PLM Integration Framework and Schema

The proposed KBE-PLM integration framework presented in Sect. 3.1 is a federated platform-independent KBE environment enabling to retrieve and integrate the information and knowledge from CAD, KBE and PLM systems in order to store and manage them in a platform independent Open KBE repository. The goal is to be able to re-use this knowledge for different KBE applications for design automation. It also ensures a reintegration of the KBE-processed knowledge within various PLM systems repositories for further re-use in downstream design activities. Considering an MBD approach for supporting KBE-PLM systems integration, we assume that most design knowledge and related design rationale (design logic) are encapsulated respectively in CAD models and in KBE systems inference engine scripts. The Open KBE repository is structured according to the KBE-PLM integration schema introduced in Sect. 3.2. This schema depicts a concept for representing in a neutral way a configured product definition, the explicit knowledge encapsulated in parameterized CAD models and the implicit knowledge related to the design intent and rationale of these models.

3.1 Architecture of the KBE-PLM Integration Framework

The KBE-PLM integration framework architecture, shown in Fig. 1, has been specified using the open enterprise modeling standard ArchiMate [28] developed by The Open Group. The framework mainly comprises a Transformation Engine including a Multi-CAD API Manager, an Open KBE Repository (the server), and an Open KBE Client. 'Open' is used in the sense of neutral. The transformation engine encompasses a set of translator libraries. These enable the transformation of native data inputs extracted from PLM system (such as PDM and ERP systems) as well as from KBE
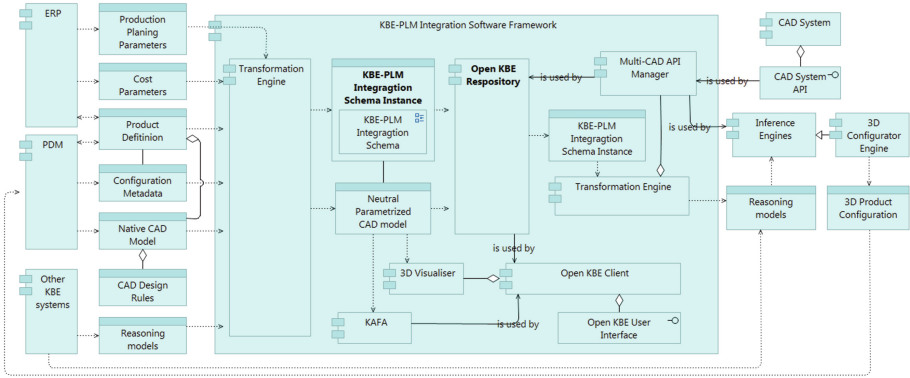
Fig. 1. KBE-PLM integration framework architecture described in ArchiMate.

systems. The nature of these inputs will depend on the addressed use case scenarios that the framework will support. The goal of the translators is to make these inputs compliant with the proposed KBE-PLM integration schema. Once the data inputs are transformed according to the schema, they can be stored in the Open KBE Repository and be interpreted and processed by the other framework components. For instance, the product structure and knowledge encapsulated in CAD parameterized assembly models can be extracted with the help of the Knowledge Acquisition and Formalization Assistant (KAFA) [14]. KAFA automatically reads the structure, components and parameters of a CAD model and generates a component-parameters design structure matrix (DSM). It extends the functionality of DSM methods by providing simple and intuitive means to enable the engineer to formalize the configuration and design logic in textual and executable forms.

The component and parameter dependencies and formalized rules defined in KAFA should enrich the Open KBE Repository according to the schema. The Open KBE client includes a user interface (UI) to access and request the Open KBE repository as well as a configurable 3D visualizer enabling to display the CAD geometry with the "fit-for-purpose" information (dimension, tolerances, parameters, rules, etc.). This UI should enable framework users to drive the required transformations for:

- translating the instantiated KBE-PLM Integration Schema and related reasoning models so that they can be processed by the right inference engine;
- driving the automation procedure in the appropriate CAD system through the use of the Multi-CAD API Manager;
- translating the outputs of the inference engine into the appropriate format so that new product definitions can be reintegrated in the various PLM systems.

3.2   The KBE-PLM Integration Schema

The KBE-PLM Integration Schema is an abstract model with system-independent and language-independent semantics. The schema defines a neutral product data and

knowledge representation enabling the exchange of parameterized CAD assembly models and related design intent and rationale. It also associates the geometric representation and rationale to a specific usage context integrating information such as configuration management data, version history and the KBE application that process the design rationale. Figure 2 shows a Unified Modeling Language (UML) class diagram of the top abstraction level of the KBE-PLM Integration Schema. The structure of the schema is aligned with the modeling principles defined in STEP integrated and generic application resources and is also based on the premise that a product artifact has a form, a particular function and a behavior. The *Product Model* is the base abstract class for all entities that constitute the definition of a product. It is a master model containing data common to all configurations and versions of a system constituent. One *Product Model* has one or more *Product Definitions* and represents an abstract entity gathering all the characteristics and design information of a *Product Model*. The three *Product Definition* sub-types which are *Functional Definition*, *Structural Definition* and *Behavioral Definition* respectively gather the product functional variables, structural design variables and behavioral features of a *Product Model*.
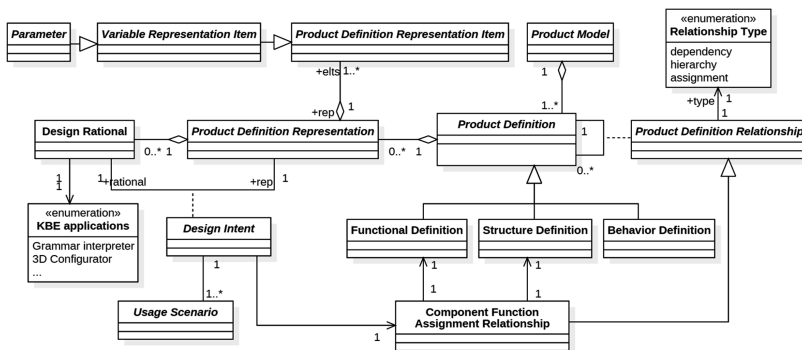

Fig. 2. KBE-PLM integration schema – top abstraction level.

A *Product Definition* consists of a set of *Product Definition Representations* (e.g. CAD models) and properties (e.g. mass and material properties). The product definition relationship entity enables to set up different types of relationship (dependency, hierarchical, etc.) so as to have traceability and consistency between the functional, structural and behavioral definition of a component. The **Design Intent** is a concept that, based on a specific usage context (defined by the *Usage Scenario* entity), enables the designer or engineer to know which product structural representation and related design rationale to use to fulfill a specific functional intent. The functional intent (defined by the *Component Function Assignment Relationship*) allows linking one *Functional Definition* to one *Structure Definition* to know what are the functions and related design requirements this structural definition intends to fulfill. Concretely, the design intent allows linking a specific *Product Definition Representation* to a specific

Design Rationale and the related *KBE application* that can process it. Figure 3 shows a more detailed level of the schema depicted in Fig. 2, focusing on instantiations of some of the abstract classes directed at:

- representing a parameterized CAD assembly model;
- representing the KBE rules through the use of the abstract class Variational Representation Item, the instances of which are the Parameters and Explicit Constraints defining the mathematical functions between parameters;
- defining the specific relationships between a geometric model and its topological elements, its design rationale, functional requirements and design parameters.

To represent an assembly model an *Assembly Component Usage* (sub-type of *Product Relationship*) describes which components are present in each parent assembly component as well as its relative *Position*. A *Structure Definition* also aggregates a set of *Properties* class which can be *Item Property* or *Shape Dependent Property*. The Structure Definition also aggregates *Structure Model(s)* which is an abstract class defining various models derived from the *Structure Definition*. In Fig. 3 below, only the *Geometric Model* is considered.
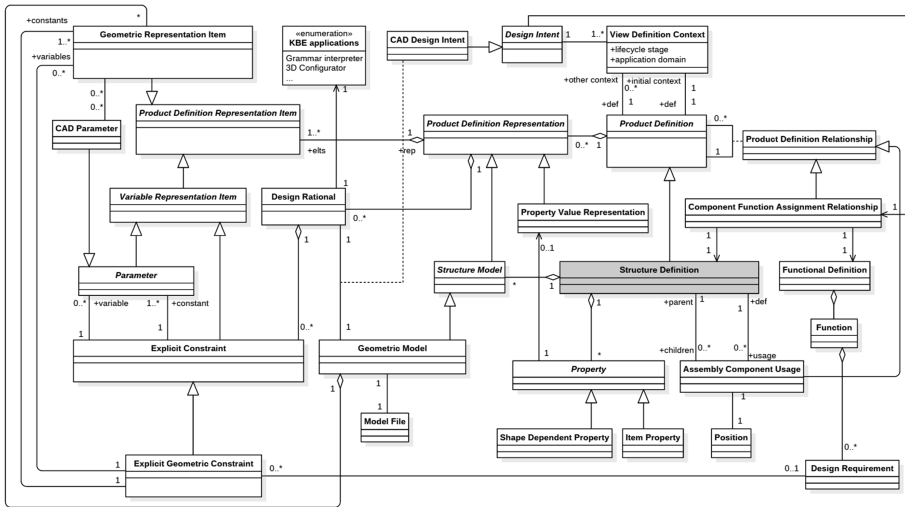
Fig. 3. KBE-PLM integration schema – CAD assembly models, CAD design intent, design rationale and parameterization.

A *Geometric Model* is an aggregation of *Geometric Representation Items* (topological element of the CAD model). In the STEP and Feature Based Parametric Modeling paradigms, the *Geometric Representation Items* are driven by parameters and constraints. In this schema the constraints (represented by the *Explicit Constraint* entity) define the dependency rules and the mathematical functions (as defined in ISO10303-50) between *Parameters*. In the above schema, there is a complex but

necessary inheritance structure to manage and link these *Geometric Representation Items* to their related parameters and constraints. This inheritance mechanism is necessary to further extend the schema and to define constraints and/or rules between other types of parameters and product definition representation items from other types of product definition representations (behavioral representation or item property representations). The *Geometric Representation Item* entity and the *Variational Representation Item* entity are direct sub-class of the *Product Definition Representation Items* (which are the constituting elements of a representation whatever its type). The *Parameter* and *Explicit Constraint* entity are both sub-types of the *Variational Representation Item* entity. The abstract *Explicit Constraint* entity links a set of constant and variable parameters. The *Explicit Geometric Constraint* entity is the "instantiable" sub-type for linking a set of constant and variable CAD model parameters that are related to the some topological elements (*Geometric Representation Item*) of the CAD model.

Finally to integrate configuration context information with the Product Definition and its Design Intent, the proposed KBE-PLM integration schema is actually a multi-layer schema as the one defined by Vosgien [21]. As shown in Fig. 4, it integrates a system definition layer and a configuration layer in order to be able to ensure the interoperability with PDM systems through standardized data exchange. A *Configuration* is a product variant (or Configuration Item in ISO10303-44), i.e. the identification of a particular variant of a product. A product variant is defined with respect to the product class (product concept in ISO10303-44), i.e. the class of similar products of which it is a member.

Concretely, Fig. 4 shows the data model enabling to build, capture and modify the various product configurations and how they are linked to their constituting elements (Assembly Component Usage of Product Definitions) through the concept of
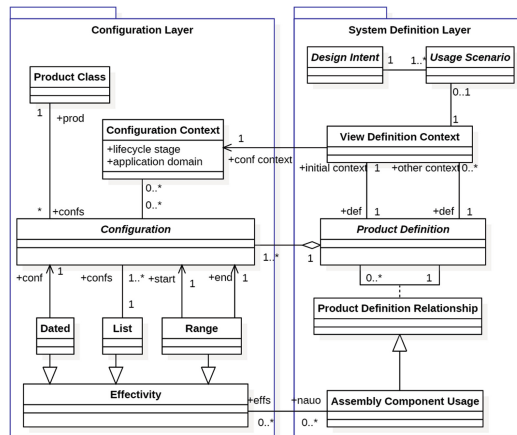

Fig. 4. KBE-PLM integration schema – integration of configuration context information with the product definition and its design intent. (adapted from [31])

"Effectivity". *Effectivity* is the key concept for linking the system Product Definitions to the appropriate product configurations in which they are valid as defined in. Instances of *Effectivity* may be applied to any *Product Definition* instances (or *Assembly Component Usage*). The *View_Definition_Context* defines the context in which the *Product Definition* of a product model is valid. A *View Definition Context* is the grouping of a *Configuration_Context* (defining the application domain(s) and the life cycle stage(s) in which a *Product Definition* is valid). In order to specify the usage and potential re-use of a representation with a design rationale in a specific configuration context, the *Design Intent* entity is linked to one *Usage Scenario* which refers to at least one *View Definition Context Definition*.

Summarizing, the core contribution of this schema is built on the *Design Intent* entity and its instantiation for CAD models (the *CAD Design Intent*) defining the proper defined relationships between a CAD model, its design rationale and its KBE usage scenario.

4   Discussion

The KBE-PLM integration framework and schema have been presented above. The schema has to be enriched with all *Geometric Representation Item* sub-types in order to be able to represent and exchange any kind of geometric model. The construction history has not been taken into account yet in the schema but will have to be addressed in future work. The feasibility of implementing such a data model will be demonstrated in future work addressing several test cases (see next section). With respect to the implementation of the framework components, we face some heavy refactoring work based on a number industry-strength prototypes of configurators and design automation systems developed by V-Research in earlier and current industrial research projects as part of COMET ProDSS [29] and AEDA K-Projects (see below). We also have a KAFA prototype from TU Wien, and the neutral CAD-model representation is part of the first author's PhD project. The Multi-CAD API Manager is already operational with bidirectional interfaces (at different maturity levels) to three major CAD systems and was developed by V-Research.

The major limitation of the presented approach is the need to develop a large set of interfaces for CAD/KBE/PLM systems to incorporate them into the framework. Moreover, any evolution in these systems will entail adapting these interfaces to the evolved systems, e.g. CAD systems and their APIs. Thus, for full-potential leverage, standardization initiatives would be important in order to involve industrialists and software vendors in the development of the standard and in the implementation and maintenance of these interfaces. That is why the proposed KBE integration schema could serve as a basis to re-launch research work targeted on answering the OMG request for proposal "KBE services for PLM". Additionally, it could be used to demonstrate the feasibility of implementing ISO 10303-108 and be able to use this part of the schema in both PLM and KBE repositories.

5   Conclusion and Way Forward

The proposed KBE-PLM integration framework presented in Sect. 3.1 is a federated platform-independent KBE environment permitting to retrieve and integrate the information and knowledge from CAD, KBE and PLM systems. The goal is to be able to re-use this knowledge for different KBE applications for design automation. It also ensures a re-integration of the KBE-processed knowledge within various PLM systems repositories for further re-use in downstream design activities. The framework encompasses a platform-independent Open KBE repository structured according to the KBE-PLM integration schema introduced in Sect. 3.2. This schema depicts a concept for representing, in a neutral way, configured product definitions, the explicit knowledge encapsulated in parameterized CAD models and the implicit knowledge related to the design intent and rationale of these models. The work presented here only represents a starting point. The following issues will be addressed in future work. As mentioned, we are currently reviewing use-case scenario candidates and will correspondingly capture industrial test case (TC) datasets. These datasets will be used to demonstrate the feasibility to instantiate the proposed schema and implement such a platform-independent KBE repository generating a data base with the data model. Then a first client prototype will be developed to be able to enrich and request this database. Hence, KAFA will be extended to automate the extraction of the knowledge from the TC datasets, and the database will be enriched automatically. The Multi-CAD API Manager CADAL (CAD Abstraction Layer) is an ongoing development effort at V-Research.

References

1. Brown, S.L., Eisenhardt, K.M.: Product development: past research, present findings, and future directions. Acad. Manage. Rev. **20**(2), 343–378 (1995)
2. 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. Adv. Eng. Inform. **26**(1), 5–15 (2012)
3. Kim, J., Pratt, M.J., Iyer, R.G., Sriram, R.D.: Standardized data exchange of CAD models with design intent. Comput.-Aided Des. **40**(7), 760–777 (2008)
4. Rachuri, S., Subrahmanian, E., Bouras, A., Fenves, S.J., Foufou, S., Sriram, R.D.: Information sharing and exchange in the context of product lifecycle management: role of standards. Comput.-Aided Des. **40**(7), 789–800 (2008)
5. Catic, A., Malmqvist, J.: Towards integration of KBE and PLM. In: Proceedings of the International Conference on Engineering Design (ICED 2007) (2007)
6. Bermell-Garcia, P., Fan, I.-S.: Practitioner requirements for integrated knowledge-based engineering in product lifecycle management. Int. J. Prod. Lifecycle Manag. **3**(1), 3–20 (2008)

7. Stokes, M.: Managing Engineering Knowledge: MOKA - Methodology for Knowledge Based Engineering Applications. Professional Engineering Publication, London (2001)
8. Schreiber, G.: Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, Cambridge (2000)
9. Curran, R., Verhagen, W.J.C., van Tooren, M.J.L., van der Laan, T.H.: A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD. Expert Syst. Appl. **37**(11), 7336–7350 (2010)
10. Hoisl, F., Shea, K., Helms, B., et al.: Towards representing, evolving and networking engineering knowledge for computational design synthesis. In: The 10th International Design Conference on Proceedings DESIGN 2008, Dubrovnik, Croatia (2008)
11. Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horváth, I., Bernard, A., Harik, R.F., Gao, W.: The evolution, challenges, and future of knowledge representation in product design systems. Comput.-Aided Des. **45**(2), 204–228 (2013)
12. Gero, J.S., Kannengiesser, U.: The situated function–behaviour–structure framework. Des. Stud. **25**(4), 373–391 (2004)
13. Quintana, V., Rivest, L., Pellerin, R., Venne, F., Kheddouci, F.: Will model-based definition replace engineering drawings throughout the product lifecycle? A global perspective from aerospace industry. Comput. Ind. **61**(5), 497–508 (2010)
14. Gerhard, D., Lutz, C.: IT-based configuration and dimensioning of customer specific products – towards a framework for implementing knowledge based design assistant systems. In: 68-6: Proceedings of the 18th International Conference on Engineering Design, ICED 2011 (2011)
15. Object Management Group: Model Driven Architecture (MDA) (2014)
16. Truyen, F.: The Fast Guide to Model Driven Architecture the Basics of Model Driven Architecture. Cephas Consulting Corporation, Tustin (2006)
17. Pablo, B.-G., Ip-Shing, F., Adrian, M. et al.: Towards the semantic interoperability between Kbe and PLM systems. Guidelines for a Decision Support Method Adapted to NPD Processes (2007)
18. ISO 10303-242: Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 242: Application Protocol: Managed Model-Based 3D Engineering (2014)
19. ISO 10303-108:2005: Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 108: Integrated Application Resource: Parameterization and Constraints for Explicit Geometric Product Models (2005)
20. Lee, H.C., Gaikwad, P.K., Adédjouma, S.A., Azehoun-Pazou, G., Tall, K., Mboup, M.L., Farssi, S.M., Diop, I., Diop, A.K., Sissoko, G. et al.: Strategies for using STEP in parametric design. Strategies **2**(4) (2013)
21. Vosgien, T.: Model-based system engineering enabling design-analysis data integration in digital design environments: application to collaborative aeronautics simulation-based design process and turbojet integration studies. Ph.D. thesis, Ecole Centrale Paris (2015)
22. Fenves, S.J., Foufou, S., Bock, C., Sriram, R.D.: CPM2: a core model for product data. J. Comput. Inf. Sci. Eng. **8**(1), 014501 (2008)
23. Finin, T., McKay, D., Fritzson, R.: An overview of KQML: a knowledge query and manipulation language. Citeseer (1992)
24. Genesereth, M.R., Fikes, R.E. et al.: Knowledge interchange format-version 3.0: reference manual (1992)
25. Kifer, M.: Rule interchange format: the framework. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 1–11. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88737-9_1

26. Sanya, I.O., Shehab, E.M.: An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry. Int. J. Prod. Res. **52**(20), 6192–6215 (2014)
27. Fan, I.-S., Bermell-Garcia, P.: International standard development for knowledge based engineering services for product lifecycle management. Concurr. Eng. **16**(4), 271–277 (2008)
28. Lankhorst, M.,M., Proper, H.,A., Jonkers, H.: The architecture of the ArchiMate language. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS/EMMSAD -2009. LNBIP, vol. 29, pp. 367–380. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01862-6_30
29. Frank, G., Entner, D., Prante, T., Khachatouri, V., Schwarz, M.: Towards a generic framework of engineering design automation for creating complex CAD models. Int. J. Adv. Syst. Meas. **7**(1, 2), 179–192 (2014)