



HAL
open science

Engineering Knowledge Extraction for Semantic Interoperability Between CAD, KBE and PLM Systems

Jullius Cho, Thomas Vosgien, Detlef Gerhard

► **To cite this version:**

Jullius Cho, Thomas Vosgien, Detlef Gerhard. Engineering Knowledge Extraction for Semantic Interoperability Between CAD, KBE and PLM Systems. 14th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2017, Seville, Spain. pp.568-579, 10.1007/978-3-319-72905-3_50 . hal-01764169

HAL Id: hal-01764169

<https://inria.hal.science/hal-01764169>

Submitted on 11 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Engineering Knowledge Extraction for Semantic Interoperability between CAD, KBE and PLM Systems

Jullius Cho¹, Thomas Vosgien², Detlef Gerhard¹

¹TU Wien, Vienna, Austria
{jullius.cho, detlef.gerhard}@tuwien.ac.at

²V-Research GmbH, Dornbirn, Austria
thomas.vosgien@v-research.at

Abstract. For the deployment of both Product Lifecycle Management (PLM) and Knowledge-Based Engineering (KBE) approaches, product and process engineering knowledge needs to be identified, acquired, formalized, processed and reused. While knowledge acquisition is still a bottleneck process, the formalized engineering knowledge is still too often encapsulated in CAD models and in KBE systems developed in vendor-specific environments. To address this issue, this paper introduces a possible solution enabling the enrichment of a CAD-KBE-PLM integration schema that provides a standardized and neutral representation of engineering knowledge for further reuse across heterogeneous CAD, KBE and PLM systems. To enrich this schema, the proposed solution combines the use of a Multi-CAD API library – which allows platform-independent and automatic extraction of engineering knowledge from CAD models into an XML-based representation – and a Knowledge Acquisition and Formalization Assistant (KAFA) which assist domain experts to formalize their procedural knowledge.

Keywords: Knowledge Acquisition, Semantic interoperability, Computer-Aided Design, KBE-PLM Integration

1 Introduction

The Product Lifecycle Management (PLM) concept aims at integrating all information produced throughout a product's lifecycle [1]. However, the multiplicity and diversity of PLM and automation enabling technologies such as Computer-Aided Design (CAD), Knowledge-Based Engineering (KBE) and IT systems do generate and consume engineering knowledge chunks which are isolated and/or locked down in various vendor proprietary applications. To support an efficient PLM approach, the isolated chunks of engineering knowledge need to be made accessible for reuse across these applications and all along the product lifecycle. In product development, CAD systems are the most widely used authoring applications. The CAD model is a container of engineering knowledge [2] holding much information that could be used to describe the structure, the functional and behavioral aspects of artifacts they represent.

Moreover, for many design applications, the CAD model of a product is the basis on which many downstream virtual analyses are performed such as finite element analysis and computer-aided manufacturing [3]. In designing product artifacts using CAD systems, experts also need to complement their design with an understanding of their product behavior as well with their design intent. Manufacturing companies tend more and more to hoard engineering knowledge (the know-what, know-why and know-how of product designs) for partial or total reuse in different contexts and projects. However, identifying, extracting, storing, transmitting and reusing engineering knowledge is still an on-going challenge when heterogeneous applications are used [4]. This is mostly because engineering knowledge is embedded into CAD models and the know-how is locked down in the minds of domain experts, making it rather difficult to extract and reuse. These barriers can be summarized as follows:

- Technological differences in CAD applications and CAD models;
- The lack of implementations of existing open standards defining the representation and exchange protocols of engineering knowledge;
- Subjective nature of experience garnered by domain experts.

These barriers warrant research in the area of data and system interoperability amongst PLM enabling technologies. There are various approaches for deploying platform-independent authoring and IT applications and for making engineering knowledge reusable. Some of these include Model-Driven Engineering (MDE), Model-Based Definition (MBD) and the use of product data standards such as ISO 10303 [5]. While the implementations of such approaches or solutions can solve some interoperability issues, they generally lack the capability to serve up all required information ensuring cohesion and traceability of product data across various domains and downstream applications. In this paper, we analyze the semantic interoperability problem and propose a possible solution to generate a common and neutral dataset to be reused across CAD, KBE and other PLM systems. To address this challenge, we propose a solution enabling the enrichment of the CAD-KBE-PLM integration schema introduced in [6] which provides a standardized and neutral representation of engineering knowledge for re-use across heterogeneous CAD, KBE and PLM systems. To enrich this schema, the proposed solution combines the use of a Multi-CAD API library (Section 3.3) – which allows platform-independent and automatic extraction of engineering knowledge from CAD models into an XML-based representation – and a Knowledge Acquisition and Formalization Assistant (KAFA) introduced by [7] which assists domain experts to formalize their procedural knowledge (Section 3.4). In a nutshell, the proposed solution is intended to capture geometric features and parameters, assembly structures and configuration rules, design intent and rationale from CAD models and domain experts and to enable the semantic interoperability of CAD, KBE and PLM enabling systems.

2 Related Work

PLM enabling technologies such as Product Data Management (PDM) are widely used as a means to store engineering data, especially CAD models and metadata. Products such as Teamcenter, ENOVIA, Windchill, etc. come to light as vendor solutions to the concept of PLM as a global repository for complete product definition. These solutions are however, locked down in vendor environments with restrictions to interoperability with other out-of-environment solutions. Some other efforts that attempted to address this challenge include the Methodology for Knowledge Based Engineering Applications (MOKA) project [8] which provides a methodology to manage engineering data, information and knowledge as well as the Model Driven Architecture (MDA) introduced by the Object Management Group (OMG) [9]. These however, have not solved the data inconsistency gap [10], [11]. Generally, engineering knowledge represents all pieces of data, information and knowledge which define the composition of a product, the intended functionality as well as the processes required to build the complete product. In [1], the authors provide various classifications of engineering knowledge such as Product Knowledge vs. Process Knowledge. While product knowledge describes what to design in a product, the process knowledge defines how to design the product. This knowledge could be found in various documents, embedded in CAD models in the form of geometry, parameters, structure or as experience garnered by domain experts such as modeling rules and logic. This paper focuses on CAD data extraction and domain expert procedural knowledge elicitation as well as their design intent.

Most CAD vendors do offer an application programming interface (API); a gateway to programmatically gain access to the functionality of their application. In extracting information from CAD models, their APIs do offer access to the internals of the respective CAD application but with varying degrees of access as seen for example in [12], [13]. There is however, no convention for generating API's and for defining which parts of the CAD application functionality they expose. Some researches such as [14] have proposed a method of extracting valuable engineering knowledge from engineering drawings. Vendor applications such as RuleStreams, Knowledge Fusion and TactonWorks are designed to aid in the extraction, management and reuse of engineering knowledge. These are however, closed systems providing no access to out-of-environment applications.

There are a good number of methods for eliciting knowledge from design experts [15]. In [16], the authors provide a comprehensive range of tools employed to elicit engineering knowledge. From experience, the most used method is the interview in its various forms. However, knowledge elicitation from experts could be flawed due to subjectivity, bias, beliefs, etc. on both domain expert and the design engineer who may be participating in the elicitation process. To solve these problems, the KAFA offers an intuitive user interface to systematically define the relationships between parts in the assembly model. This is also enhanced by the Multi-CAD API which automatically extracts some information from CAD models, providing some kind of semi-automation.

No discussion on neutral product data formats is worthy without mention of the initiative of ISO 10303 known as STEP standard. STEP has been particularly successful in its use to exchange geometry and CAD meta-data between CAD applications as well as basic PDM meta-data and assembly product structures through its Application Protocol AP242 which is a merge of AP203 and AP214. Even though many other aspects of engineering knowledge are defined in the various STEP parts and protocols, they have not yet been implemented by vendors. Making use of the groundwork set in STEP, in [6], we have proposed a PLM-KBE integration schema combining several STEP parts and other neutral product data models. 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. In [4], the authors provide a study for a neutral format to exchange and reuse rule-based procedural knowledge across different KBE applications: the Rules Interchange Format (RIF). However, there is still a lack of standardized methods to extract and neutrally formalize knowledge from CAD and KBE models. By introducing the Multi-CAD API, this work aims at finding an integrated solution which will cover major CAD applications thereby providing users a bigger solution space.

3 Extracting, Formalizing and Reusing Design Knowledge for Design Automation Applications

In this section, we describe a method to capture explicit and tacit engineering knowledge for reuse in product development. To set the stage for this, we first define an integrated use case scenario, positioning our proposed solutions and defining what actually is meant by “engineering knowledge”. We then delve into extracting, formalizing and representing engineering knowledge for reuse across PLM enabling technologies.

3.1 Integrated use case scenario

Fig. 1 illustrates an integrated use case scenario portraying the identification, acquisition/extraction, formalization, storage and reuse of engineering knowledge. The aim is to extract engineering knowledge from parametric CAD models and reuse this knowledge in generating a configuration model. In the addressed example, the used CAD system is SolidWorks and the used KBE configurator is TactonWorks which is a SolidWorks add-on. The idea is to be able to reproduce the same scenario and/or reuse the same mechanisms and the extracted knowledge with other CAD, KBE and PLM systems.

The Multi-CAD API manager parses a CAD model and extracts engineering knowledge from it. This knowledge is used to generate the KAFA matrix. The KAFA then provides an intuitive GUI for the design engineer and the domain expert to systematically define the parameters, relationships and rules which are beyond a CAD configuration model. The output of the Multi-CAD API manager and the KAFA are

then used to enrich the KBE-PLM integration schema proposed in [6] and which provides a structured and neutral data set for semantic interoperability between PLM enabling applications. This schema can then be parsed and a configuration model generated. Particularly important for the configuration model is the general structure of the model, the relationship between the parts, the parameters and the functions that drive the parameters leading to different configurations.

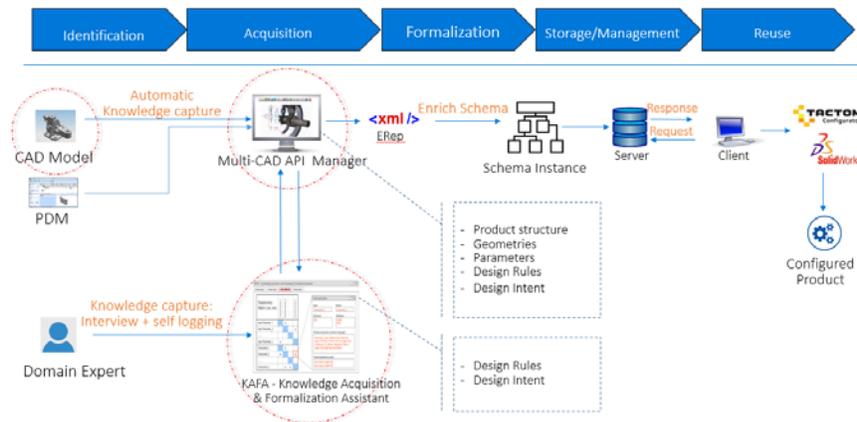


Fig. 1. Use-case scenario: acquiring, formalizing and re-using knowledge for 3D configurators

3.2 CAD data representation

Fig. 2 shows a simple tree structure portraying the type of information – structure, geometry, and topology, rules etc. – conveyed through CAD models. For this data or knowledge to be reused in other parts or processes of the product’s lifecycle, it needs to be identified, extracted and harnessed for seamless access.

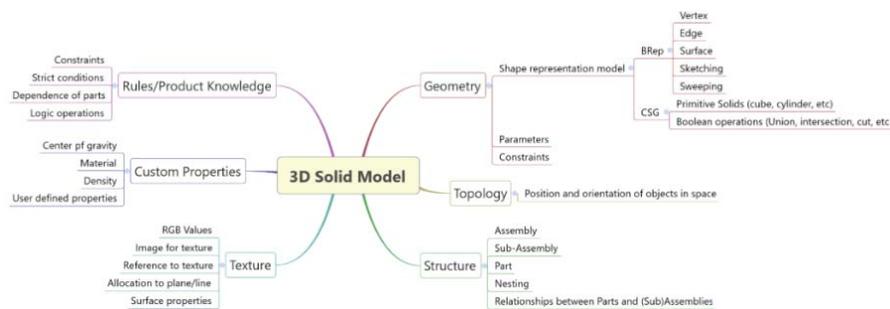


Fig. 2. Type of information embedded in CAD Models

In order to generically represent the structure, geometry, parameters and other information pertaining to CAD models without a commitment to an underlying core

modeler, we adapt the Editable Representation (ERep) proposed by [17]. An excerpt of the Domain Specific Language (DSL) for this representation is shown in **Fig. 3**.

```

1 <assembly> ::= ASSEMBLY
2             <name>
3             <stamp>
4             <global_info>
5             <parts_list>
6             END_ASSEMBLY
7
8 <global_info> ::= GLOBAL UNITS
9             <unit>
10
11 <stamp> ::= integer
12
13 <unit> ::= mm | cm | m | in | ft
14
15 <matching_rules> ::= <rule>
16                   | <matching_rules>; <rule>
17
18 <rule> ::= <mating>; <aligningn>
19
20 <mating> ::= MATE <geo_pair>
21           | MATE OFFSETv<exp><geo_pair>
22           | ALIGN <geo_pair>
23           | ALIGN OFFSET <exp><geo_pair>
24
25 <geo_pair> ::= <point> <axis> <stamp>
26
27 <part> ::= PART <name> <stamp>
28         <global_info>
29         <feature_list>
30         END_PART
31

```

Fig. 3. Excerpt of Domain Specific Language for CAD data representation adapted from [17]

3.3 Multi-CAD API for automatic knowledge extraction

The main idea behind the Multi-CAD API is to use the object-oriented programming paradigm (OOP) and build a core abstraction layer which would be implemented by different CAD applications using their respective API's as shown in **Fig. 4**. The core application or core abstraction layer defines functionality through interfaces [18], applying the principle of loose coupling [19] where the implementation of any of the interface classes could be changed without having to change the core interface class itself. These interfaces are implemented by requesting data from selected CAD applications using the respective API's that they expose, but the domain logic is retained by the core abstraction layer.

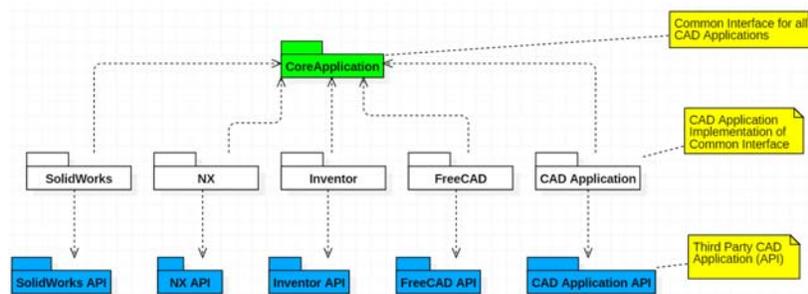


Fig. 4. Modular architecture for Multi-CAD API

Fig. 5 shows a simplified implementation of the Core Application with SolidWorks.

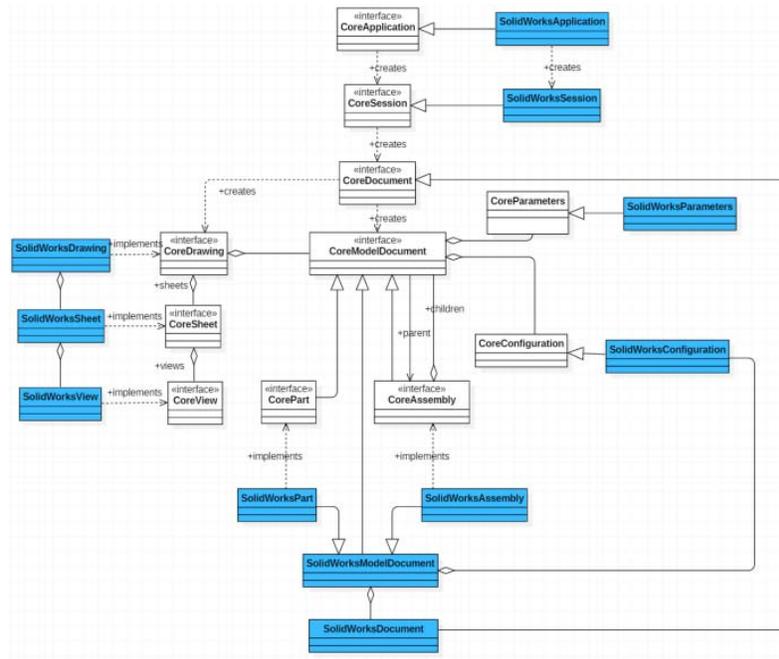


Fig. 5. Core Application implementation in SolidWorks

The core system will therefore not know which CAD application is actually in use. In this way the Multi-CAD API is independent of any CAD application for its logic and functionality. An objective of the Core Application is to make it completely generic and not dependent on any CAD solution, while preserving the accessing methods so they can be used for any similar project. The caveat would be the reliance on a dependency injection mechanism that allows the Core Application to load the required CAD data. The object model of the Core Applications consists of a Core Document which can be either a Drawing Document or a Model Document. The Model Document is either a Part or an Assembly Document. As the names indicate, the Drawing Document, Part Document and Assembly Document are object oriented abstractions which represent drawings, parts and assemblies respectively in CAD applications. Model Documents can be represented in different configurations. The Core Document also manages the parameters used in Drawing, Part and Assembly classes.

In Fig. 6, a simple 4-legged table modeled in SolidWorks was automatically parsed by the Multi-CAD API and relevant information extracted. The result is a well formed XML document containing information pertaining to the parts and assemblies. This includes the structure, the parameters at part and assembly levels, the mates and relationships between the parts in the assembly structure, etc. This XML document can now be parsed and the results used to generate an extended DSM (see Fig. 7) for elic-

iting more and relevant design intent and design rationale from a domain expert. The knowledge then has to be prepared or formalized for re-use.

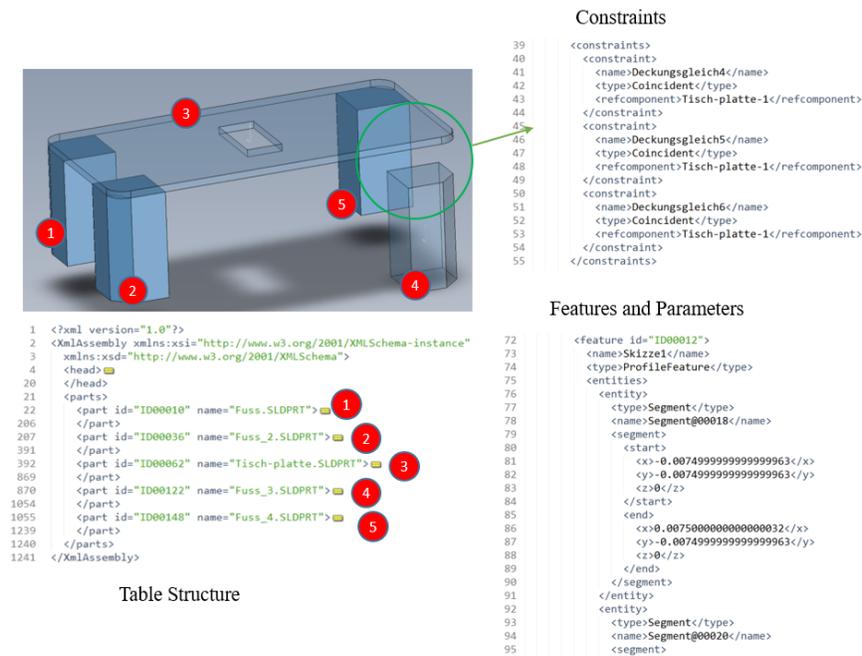


Fig. 6. Core Application implementation in SolidWorks

3.4 Knowledge Acquisition and Formalization Assistant

The main aim of the Knowledge Acquisition and Formalization Assistant as introduced by [7] is to help in capturing engineering knowledge from domain experts. One component of KAFA is a component-component parameters relationship matrix which graphically projects the parts and parameters of an assembly model in relationship to each other. The domain expert and the engineer then have an intuitive interface – a design structure matrix – to systematically define the relationships between parts in the assembly model. While the generation of the KAFA was more of a manual process, we have automated the generation process with the support of the Multi-CAD API. All components and parameters are automatically extracted from a CAD model dataset and the KAFA interface is automatically built. Moreover, KAFA has been extended with a Component-Parameter/Requirement-Function matrix to capture the design intent in a set of functional and non-functional requirements. After the design engineer and the domain expert have discussed and defined the relationship between the parts, the resulting knowledge base is then used to enrich the CAD-KBE-PLM schema.

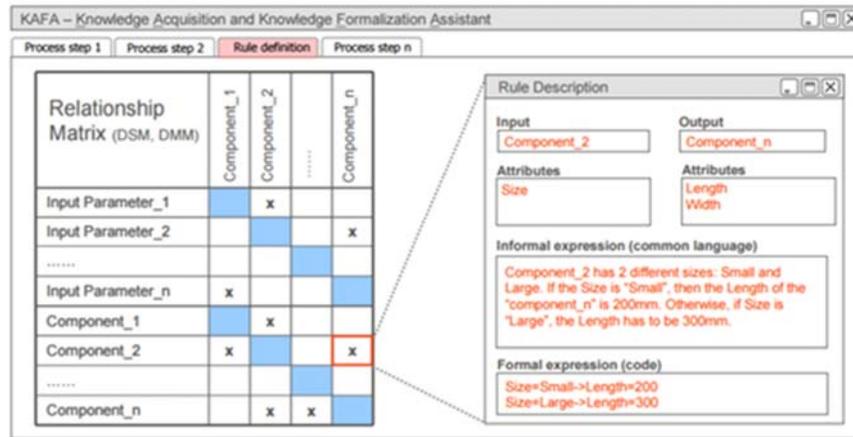


Fig. 7. Concept View of Kafa from [18]

3.5 CAD-KBE-PLM Integration

Fig. 8 shows the CAD-KBE-PLM integration schema introduced in [6]. These highlighted areas indicate some classes that will be enriched by the Multi-CAD API manager (green) and the Kafa (blue) respectively. The Multi-CAD API manager provides general information describing the structure. While the component-component matrix of the Kafa provides information about the relationships between the components, the design intent can be captured using the component/parameter – requirement matrix.

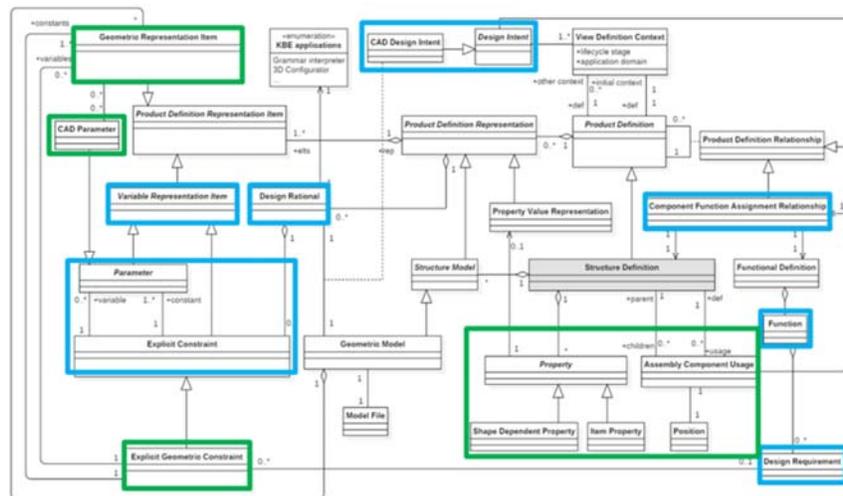


Fig. 8. CAD-KBE-PLM Integration Schema from [6]

4 Discussion

The goal of this research work is to demonstrate the feasibility of extracting relevant engineering knowledge from CAD models as well as the design intent of the engineer on the one hand, and to enable the exchange and re-use of this knowledge across heterogeneous CAD, KBE and PLM systems on the other hand.

Using OOP principles, the Multi-CAD API manager automatically extracts relevant knowledge embedded in CAD models. While the Multi-CAD API is dependent on CAD applications for data structures but is independent of any CAD application for its logic and functionality. The full capability of such a Multi-CAD API can however, only be achieved if the CAD vendors expose the required functions in their API's. As mentioned in [6], another limitation of the Multi-CAD API manager approach is the need to develop different interfaces for the different systems to be integrated. Moreover, current CAD systems provide knowledge-based capabilities where modeling rules can be defined and re-used by the user. Such objects are part of the data structure of the commercial systems which is considered as strategic intellectual property of their solutions. Although the solution and schema proposed in this paper aim at overcoming this barrier, the potential implementation of such a solution in industrial environments will depend on the willingness of these software vendors to implement and maintain the required interfaces.

For the elicitation and formalization of design intent and rationale from domain experts, the proposed KAFA proposes a formalization of component-parameters relationships (modeling rules and constraints) on the one hand and Component-Parameter/Functional Requirements relationships on the other hand. The rules defining these relationships are expressed in a natural language and transformed in executable code. One limitation is that, depending on the complexity of the product knowledge to formalize, the dataset could be very large and lead to conflicting rules. A mechanism has to be put in place to check redundancy and inconsistency thereby supporting the elicitation and formalization of procedural knowledge from the domain expert.

5 Conclusion

The investigations reported in this paper comprise an attempt at developing a solution to generate a common and neutral dataset aiming at tackling the semantic interoperability issue that prevents the efficient exchange of engineering data between CAD, KBE and PLM systems. The target source of product knowledge is the CAD model on the one hand, and the domain experts' know-how on the other hand. The ideas, results and related works presented in this paper lead to the general conclusion that the reuse of engineering data, information and knowledge across CAD, KBE and PLM enabling applications is still a major challenge and open issue in engineering design. This is mainly due to the inexistence of established standardized methods or approaches for automatically extracting and formalizing this engineering knowledge in a platform-independent and neutral standardized way. The main identified short-

comings are the lack of a suitable neutral knowledge representations with well-defined syntax, axioms, and semantics to be shared across multiple platforms and to enable interoperability [20]. This paper argues that when a structured approach towards integrating CAD, KBE and PLM enabling applications with product development is considered, consistent representations of CAD geometric features, modeling parameters, rules and constraints as well the related design intent and rationale are needed. The challenges addressed in this research work is first to be able to extract and formalized all this information from the available sources of knowledge (CAD models and domain experts) and to be able to exchange and re-use it across heterogeneous CAD, KBE and PLM applications. The Multi-CAD API demonstrates a possibility of unlocking engineering knowledge embedded in CAD models independently from the used CAD system. The extracted CAD features are then used by domain experts in order to formalize the above mentioned design intent and rationale with the support of the proposed KAFA.

The next step will be first to effectively use the output of the Multi-CAD API manager and the KAFA to enrich the CAD, KBE, PLM integration schema previously introduced in [6]. This should also be accompanied by an appropriate GUI to ease the interaction with user. The use case scenario introduced in this paper will have to be extended to include KBE and PLM data exchange steps as well as automatic validation and reporting mechanisms for checking the compliance of the extracted and formalized knowledge to the PLM-KBE integration schema.

Acknowledgments. This work was supported by the K-Project ‘Advanced Engineering Design Automation’ (AEDA) that is financed under the COMET (COMpetence centers for Excellent Technologies) funding scheme of the Austrian Research Promotion Agency.

References

1. Chandrasegaran SK, Ramani K, Sriram RD, et al (2013) The evolution, challenges, and future of knowledge representation in product design systems. *Comput-Aided Des* 45:204–228.
2. Badin J, Chamoret D, Gomes S, Monticolo D (2011) Knowledge configuration management for product design and numerical simulation. *Proc. 18th Int. Conf. Eng. Des. ICED 11 Impacting Soc. Eng. Des. Vol 6*.
3. Gerbino S, Brondi A, others (2004) Interoperability issues among cad systems: a benchmarking study of 7 commercial mcad software. *32 Proc. Des. 2004 8th Int. Des. Conf. Dubrov. Croat*.
4. Colombo G, Pugliese D, Klein P, Lutzemnberger J (2014) A study for neutral format to exchange and reuse engineering knowledge in KBE applications. In: *Eng. Technol. Innov. ICE 2014 Int. ICE Conf. On. IEEE*, pp 1–10.
5. Bondar S, Shammaa A, Stjepandić J, Tashiro K (2015) Advances in Parameterized CAD Feature Translation. In *ISPE CE*, pp. 615-624.
6. Cho J, Vosgien T, Prante T, Gerhard D (2016) KBE-PLM Integration Schema for Engineering Knowledge Re-use and Design Automation”, 13th IFIP Inter-

- national Conference on Product Lifecycle Management (PLM16), Columbia, SC (USA).
7. Gerhard D, Christoph L (2011) IT-based configuration and dimensioning of customer specific products—towards a framework for implementing knowledge based design assistant systems. 68-6 Proc. 18th Int. Conf. Eng. Des. ICED 11 Impacting Soc. Eng. Des. Vol 6.
 8. Stokes M, MOKA Consortium (2001) Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications. Professional Engineering Publ.
 9. OMG (2006) KBE Services for PLM.
 10. Barbau R, Krma S, Rachuri S, et al (2012) OntoSTEP: Enriching product model data using ontologies. *Comput-Aided Des* 44:575–590.
 11. Fowler J (1995) STEP for data management, exchange and sharing. Ta, Technology Appraisals Ltd, Twickenham, UK.
 12. Liu M, Ma J, Lin L, et al (2014) Intelligent assembly system for mechanical products and key technology based on internet of things. *J Intell Manuf* 1–29.
 13. Mathew A, Rao CSP (2010) A CAD system for extraction of mating features in an assembly. *Assem Autom* 30:142–146.
 14. Amran MFM, Sulaiman R, Kahar S, et al (2012) A Proposed Information Extraction Technique in Engineering Drawing for Reuse Design. *World Acad Sci Eng Technol Int J Mech Aerosp Ind Mechatron Manuf Eng* 6:441–445.
 15. Verhagen WJC, Bermell-Garcia P, van Dijk REC, Curran R (2012) A critical review of Knowledge-Based Engineering: An identification of research challenges. *Adv Eng Inform* 26:5–15.
 16. Burge JE (2001) Knowledge elicitation tool classification. *Artif. Intell. Res. Group Worcest. Polytech. Inst.*
 17. Hoffmann CM, Juan R (1992) Erep An Editable High-Level Representation for Geometric Design and Analysis.
 18. Goyal G, Sachin P (2012) Importance of Inheritance and Interface in OOP Paradigm Measure through Coupling Metrics. *Int. J. Appl. Inf. Syst.* 4:
 19. Leymann F (2016) Loose Coupling and Architectural Implications [Powerpoint slides]. Retrieved from <http://esocc2016.eu/wp-content/uploads/2016/04/Leymann-Keynote-ESOCC-2016.pdf>
 20. Trehan V, Chapman C, Raju P (2015) Informal and formal modelling of engineering processes for design automation using knowledge based engineering. *J Zhejiang Univ Sci A* 16:706–723.