Advanced Engineering Informatics 26 (2012) 5–15

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

A critical review of Knowledge-Based Engineering: An identification of research challenges

Wim J.C. Verhagen [a,*], Pablo Bermell-Garcia [b,1], Reinier E.C. van Dijk [c,2], Richard Curran [a,3]

[a] Air Transport & Operations, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft, The Netherlands
[b] Intelligent and Semantic Systems, EADS IW Technical Capability Center 5 (Simulation, Environment of Systems, IT & Systems Engineering), Golf Course Lane, BS997AR, Filton, Bristol, UK
[c] Systems Engineering & Aircraft Design, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft, The Netherlands

ARTICLE INFO

Article history:
Received 18 January 2011
Accepted 23 June 2011
Available online 20 July 2011

Keywords:
Knowledge-Based Engineering
Review
Knowledge
Knowledge Engineering
KBE
Intelligent systems

ABSTRACT

Knowledge-Based Engineering (KBE) is a research field that studies methodologies and technologies for capture and re-use of product and process engineering knowledge. The objective of KBE is to reduce time and cost of product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and re-using design knowledge. Published research on KBE is not very extensive and also quite dispersed; this paper is an effort to collect and review existing literature on KBE. A total of 50 research contributions have been analysed. The main objectives of this analysis are to identify the theoretical foundations of KBE and to identify research issues within KBE, pointing out the challenges and pitfalls that currently prohibit a wider adoption of KBE while suggesting avenues for further research. Key findings include (a) the necessity for improved methodological support and adherence, (b) better transparency and traceability of knowledge, (c) the necessity for a quantitative framework to assess the viability and success of KBE development and implementation projects, and (d) the opportunity to move towards mass customization approaches through distributed deployment of KBE in the extended enterprise.

1. Introduction

Increasingly competitive and demanding markets are forcing companies to search for means to decrease time and costs for new product development, while satisfying customer requirements and maintaining design quality. Companies are increasingly moving towards mass customization: 'the production of goods to meet individual customer's needs with near mass production efficiency' [1]. This requires the capability to rapidly and collaboratively [2] design and produce a large number of product variants, often based on modular design principles. In the design domain, one of the technologies that support rapid, modular design is Knowledge-Based Engineering (KBE); this technology has indeed been used to support mass customization [3–5]. KBE has its roots in the 1980s, when the application of Artificial Intelligence (AI) and Knowledge Engineering techniques in Computer Aided Design (CAD) became known as Knowledge-Based Engineering [6,7]. KBE is applied to automate repetitive design tasks and achieve significant design time savings, enabling designers to explore a larger part of the design space during the various design phases.

Knowledge-Based Engineering has held great promise since its first applications. Despite this promise and associated reported advantages to its adoption, KBE has to date not achieved a convincing breakthrough, apart from major aerospace and automotive companies. The reasons for this are varied and complex. Notably, the KBE research field is still in development, with methodological and technological considerations constantly evolving [6,8–11]. Though there are KBE research papers available from various application domains, very few of these papers reflect on what is known about the KBE research field as a whole. At present there is little consensus on what constitutes the theoretical foundations of the research field. Given this state of affairs, this paper is an effort to collect and review existing literature on KBE, with the main objectives of identifying the theoretical foundations of KBE and identifying research issues within KBE.

To achieve these objectives, a critical review of 50 suitable research papers has been performed. This literature sample will be discussed in more detail in Section 2. The findings resulting from the analysis of this sample will be discussed in two steps. First, the theoretical foundations of KBE will be discussed to arrive

* Corresponding author. Tel.: +31 0 15 278 8190; fax: +31 0 15 278 2062.
E-mail addresses: w.j.c.verhagen@tudelft.nl (W.J.C. Verhagen), Pablo.Bermell-Garcia@eads.com (P. Bermell-Garcia), R.E.C.vanDijk@tudelft.nl (R.E.C. van Dijk), R.Curran@tudelft.nl (R. Curran).
[1] Tel.: +44 0 1633 714479.
[2] Tel.: +31 0 15 278 5332; fax: +31 0 15 278 9564.
[3] Tel.: +31 0 15 278 1513; fax: +31 0 15 278 2062.

1474-0346/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2011.06.004

at a more consolidated view of this research field; this includes a definition of terms, a discussion of the rationale behind KBE, evaluation of KBE methodologies and identification of current shortcomings. This first step is followed by discussion of a number of research issues and challenges in Section 4. As a final step, some conclusions are drawn.

2. Review process

The literature on Knowledge-Based Engineering contains a small number of conceptual papers that endeavour to establish theoretical foundations for KBE. These papers originate from different perspectives: efforts at establishing a methodological base for KBE [9,11,12], efforts at clarifying the technical aspects of KBE [6,13,14], and efforts at connecting KBE with different domains, such as the integration of KBE with Product Lifecycle Management (PLM) [8,15,16]. However, a consolidated view on the theoretical foundations of KBE is still lacking, as is a critical review of existing literature. The objective of this paper is to provide such a review. In the following section, the selection, classification and review criteria for the literature sample are presented, followed by a brief discussion of the review sample characteristics. Fig. 1 summarizes the selection, classification and review process, which will be addressed in more detail in Sections 2.1 and 2.2.

2.1. Selection and classification

A two-stage selection process has been adopted. It uses a small set of selection criteria to arrive at a literature sample, and several categorization criteria have been identified to properly categorize this sample (see also Fig. 1).

An initial search on a small variety of keywords within established search engines (SCOPUS, ISI Web of Knowledge, and Google Scholar) has been performed. The keywords and keystrings were "Knowledge-Based Engineering" (with and without hyphen), "KBE", "engineering knowledge management" and "design automation". The latter two terms yielded a very high quantity of results; consequently, only journal papers with some impact (more than 0 citations) for "design automation" and "engineering knowledge management" were included into a first selection round. This round involved a parsing of all search results through title and abstract. A second selection round was performed by the authors as a group, based upon full and independent reading of papers. Two selection criteria were applied at this point: for positive selection, papers had to be in or closely related to the engineering domain (for instance, a considerable number of papers from the information sciences primarily related to ontological modeling were removed) and duplicate recordings of the same work were removed (some journal papers proved to be based upon earlier conference papers, to the point of being virtually identical; the journal versions were included into the literature sample). The literature sample that has resulted from the selection process is described in more detail in Section 2.2.

With respect to classification of the literature sample, the authors have chosen to refrain from explicitly applying any specific philosophical paradigm (e.g. positivist versus interpretivist [17]) for classification and analysis of papers while reviewing KBE literature, even though this is a widely accepted review approach (see e.g. Guo and Sheffield [17] and Rezgui et al. [18]). The added value of a classification of research according to philosophical paradigms is limited in the case of KBE literature, as the volume of KBE literature is limited, which in turn would take away from the significance and validity of any philosophically categorized findings. Moreover, the practical utility of selection and classification criteria is considered to be most important. This fits with the authors' observation that KBE literature is firmly rooted in the larger category of design science [17,19], where the establishment of scientifically-based methodologies for aiding and performing

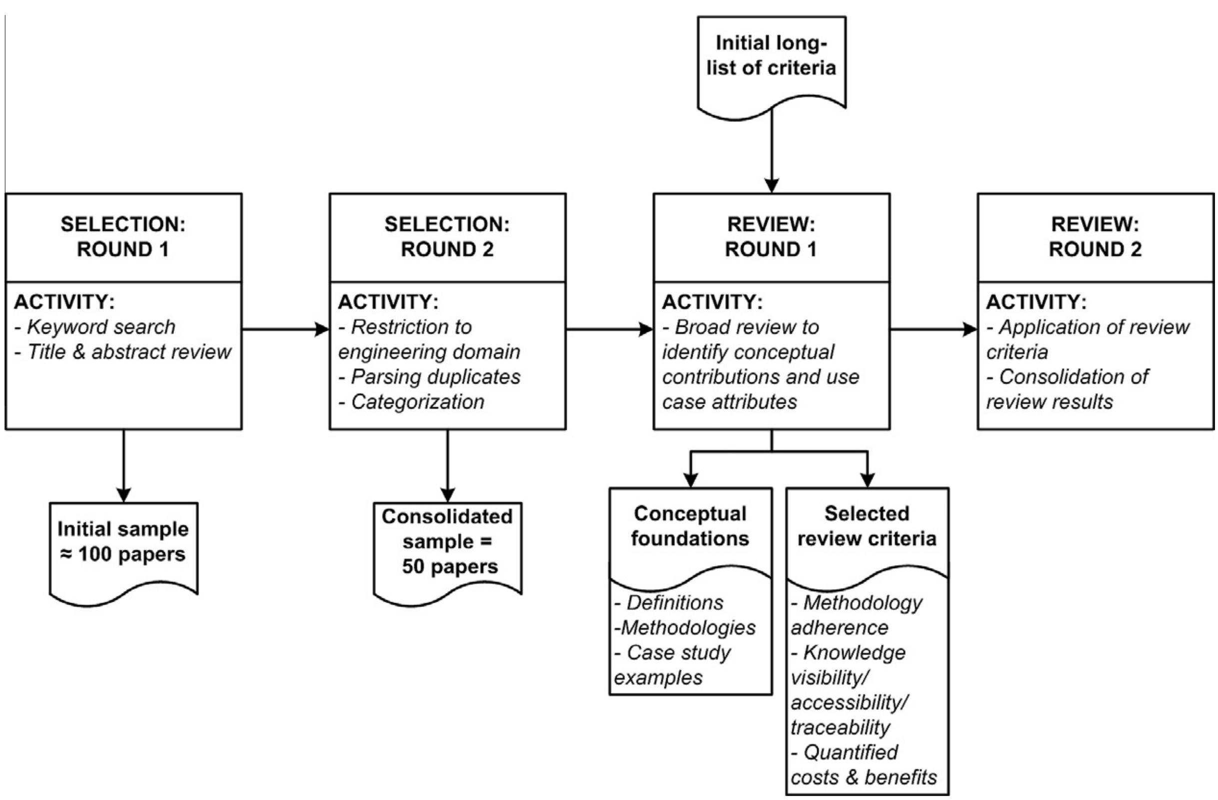

Fig. 1. Selection, classification and review process.

Table 1
Main aspects of contribution types.

Type of contribution	Main aspects
Methodology article	• New KBE methodology is proposed with respect to KBE development or KBE utilization • Update of or improvements to an existing methodology are proposed
Review article	• Review of KBE research field • Review of KBE in relation with other research fields (e.g. PLM)
Case study article	• Focus on application development, features, processes and benefits • Validation examples of KBE applications.

design in practice followed by validation, for instance through case studies, is prevalent.

The principal categorization criterion for the literature sample is the type of contribution. Three types are identified: methodology articles, review articles and case studies. To distinguish between the three types, the main aspects per type are considered – see Table 1.

In practice, research papers often do not neatly fall in one category; a paper can be classified according to multiple contributions. For instance, a paper by Ammar-Khodja et al. [20] contains review elements, a major methodological improvement element and some validation effort. In such cases, the stated intent of the paper's authors guides categorization. For the paper by Ammar-Khodja et al. [20], this points towards a categorization as 'Methodology article'.

2.2. Sample review process

During the first selection round of the two-stage selection process previously described, the literature sample was brought down from an initial 500 applicable papers to some hundred papers. After the second selection round (selection for engineering domain; removal of duplicate recordings), the final literature sample for the review of Knowledge-Based Engineering was arrived at. This sample consists of 50 contributions [3–16,20–55]. The sample contains 27 journal papers, 15 conference papers, four dissertations, one handbook, two technical reports and one magazine article. The time of publication spans a period from 1998 to 2010. Of the 50 contributions, 37 contributions are case studies [3–5,7,13,21–33,36–43,45–55], eight contributions are methodology articles [9–12,14,20,34,35] and five contributions are review articles [6,8,15,16,44].

After the final sample was arrived at, the 50 papers were reviewed twice. Before starting the first review round, a list of candidate review criteria was composed. The first review was aimed at identification of the conceptual foundations of KBE and at consolidation of the review criteria by selection of the most salient criteria, i.e. those reflecting the most common issues that were encountered in the first review of the 50 papers. After the first review round, a short list of review criteria was arrived at. It contained methodology adherence, knowledge visibility, accessibility and traceability, and quantification of costs and benefits as criteria. The second review round was aimed at application of these review criteria to the review sample, resulting in quantified insights into current KBE developments, problems and research issues.

In Section 3, the findings from the review process will be discussed in detail. Subsequently Section 4 will discuss the resulting research challenges.

3. Consolidated review findings: conceptual foundations & identified shortcomings

In this section, the conceptual foundations of Knowledge-Based Engineering will be evaluated based upon review of the literature sample. This evaluation includes a concept definition, the rationale behind KBE adoption and a review of existing methodologies; the evaluation excludes discussion of the technological aspects of KBE, as this is a very specific topic that merits a dedicated contribution (see e.g. Cooper & La Rocca [6]). This section is completed by an identification of current shortcomings of KBE, which is subsequently connected to a discussion of research issues in Section 4.

3.1. Concept definitions

Knowledge-Based Engineering is subject to a variety of interpretations and related definitions [6,10,21,23,25,51]. As Ammar-Khodja et al. [20] note: 'In reality, there is no unambiguous definition of KBE. However, most of them are similar'. To see whether this assertion is true, a number of definitions are presented in chronological order.

An early definition originates in Sainter et al. [10], who asserted in 2000 that 'a KBE system can be regarded as a type of knowledge-based system that performs tasks related to engineering. KBE systems do not express designs with specific data instances, as ordinary CAD systems do, but with sets of rules that enable the design to apply to large classes of similar parts'. In this, Sainter et al. refers to the generative aspect of KBE systems, but no further KBE aspects are taken into account. A more involved definition can be found in Chapman & Pinfold [25], who indicate that 'KBE represents an evolutionary step in Computer-Aided Engineering (CAE) and is an engineering method that represents a merging of Object-Oriented Programming (OOP), artificial intelligence (AI) and Computer-Aided Design (CAD) technologies, giving benefit to customized or variant design automation solutions'. A comparable definition is that of Cooper et al. [3], who state that KBE is 'a particular type of knowledge-based system that is based upon an object-oriented programming language and is tightly integrated with a geometric modeling tool'. KBE enables 'generative modeling' that allows the near instantaneous generation of new design data and 'integrated modeling' that provides the means to automatically create views to support a wide range of product development activities [3].

Bermell-García & Fan [23] maintain that KBE is 'a special type of KBS with a particular focus on product engineering design activities such as analysis, manufacturing, production planning, cost estimation and even sales. The technology provides a high degree of design integration and automation in well defined and complex design tasks'. Cooper & La Rocca [6] and Van der Laan [51] state that KBE can be defined as 'the use of dedicated software language tools (i.e. KBE systems) in order to capture and re-use product and process engineering knowledge in a convenient and maintainable fashion. The ultimate objective of KBE is to reduce the time and cost of product development by automating repetitive, non-creative design tasks and by supporting multidisciplinary integration in the conceptual phase of the design process and beyond'. Finally, Baxter et al. [21] define KBE in the following manner: 'Knowledge-Based Engineering is generally regarded as an umbrella term describing the application of knowledge to automate or assist in the engineering task'.

The concept of knowledge lies at the heart of KBE; it has received much attention, literally throughout the centuries; for a solid discussion of knowledge within the context of knowledge-based systems, please refer to Alavi & Leidner [56] and Schreiber et al. [57].

Having reviewed the KBE definitions, the statement of Ammar-Khodja et al. [20] is indeed appropriate: the definitions available in literature are quite similar. In retrospect, an evolution from narrow to wide definitions can be perceived. Older definitions are more narrow and technology-oriented; for instance, the notion of KBE as a combination of CAD and AI techniques [10,25]. More recent definitions of KBE [6,23,51] are wider and less restrictive; they for instance do not contain the geometry focus that often seems to constrain KBE applicability. Instead, newer definitions focus on the automation of repetitive engineering tasks while capturing, retaining and re-using associated knowledge. KBE sets itself apart from more conventional automation approaches by including this focus on knowledge capture, retention and re-use. This view of KBE will be revisited in Section 4.2: Identification of research issues.

3.2. The rationale for Knowledge-Based Engineering

There are a number of strong arguments for adopting Knowledge-Based Engineering. There are also compelling reasons not to pursue KBE. In this section, both viewpoints will be discussed.

Most of the arguments in favour of adopting KBE have their roots in the opportunities offered by the rationalization and automation of design in the conceptual and preliminary design phases. A well-recognized feature of design is that a large percentage of the product's life cycle cost is committed by the end of the preliminary design phase (Fig. 1). Estimates for committed cost range between 60% and 80% [24,27,45,58]. The decisions that drive cost are made based on trade-offs. It would be best if the design envelope would be explored to the fullest extent possible before making decisions with a high impact on committed costs; currently, the knowledge upon which a designer can base decisions in the early design phase is limited. However, the time-intensive nature of many routine design tasks – up to 80% of design time is spent on routine tasks [11] – commonly prohibits a full exploration of options. This is illustrated in Fig. 2. Furthermore, this figure expresses that KBE has the potential to bring knowledge forward in the design process.

A major advantage in adopting KBE is highlighted in Fig. 3. As the definition of KBE states, one of the hallmarks of the KBE approach is to automate repetitive, non-creative design tasks. Not only does automation permit significant time and cost savings, it also frees up time for creativity [3,51], which allows exploration of a larger part of the design envelope. This is helped by another advantage of KBE: it enables knowledge re-use (up to a degree – see also Section 4). As Baxter et al. [21] note, 'around 20% of the designer's time is spent searching for and absorbing information', and '40% of all design information requirements are currently met by personal stores, even though more suitable information may be available from other sources' [21]. This implies that design information and knowledge is not represented in a shared and easily accessible knowledge base. Clearly, in such cases knowledge re-use guided by an established KBE framework may save considerable time and effort.

Another major advantage of KBE is its integrated modelling approach, where a single, central representation of the required knowledge is maintained. This allows the ability to leverage upon a shared knowledge base and offer domain-specific views of a design problem, for instance a manufacturing product view (Van der Laan [51]). This helps in the understanding and subsequent analysis efforts for involved design engineers.

Other KBE benefits are associated with the activities in the knowledge management phase of KBE development, in which the elicitation of knowledge has benefits in itself [33]. First of all, implementation requires reflection on and analysis of engineering activities. When the business case is pursued, knowledge elicitation enforces an interdisciplinary exchange of information, knowledge sharing and the establishment of collaboration networks. Also, this elicitation phase allows documentation of engineering best practices. Furthermore, the process of implementing and deploying KBE has other beneficial side effects like standardizing terminologies, clarifying procedures and identifying engineering decisions.

Successful efforts to deploy and benefit from KBE have been made in various domains. In the automative domain for example, Chapman & Pinfold [25] describe the development of a multidisciplinary KBE system that is able to rapidly generate car designs. In the aerospace domain, efforts by La Rocca & Van Tooren [41] as well as Corallo et al. [29] describe KBE systems that automate design effort for respectively Blended Wing Bodies, and low pressure turbines. KBE systems are also developed for the design of

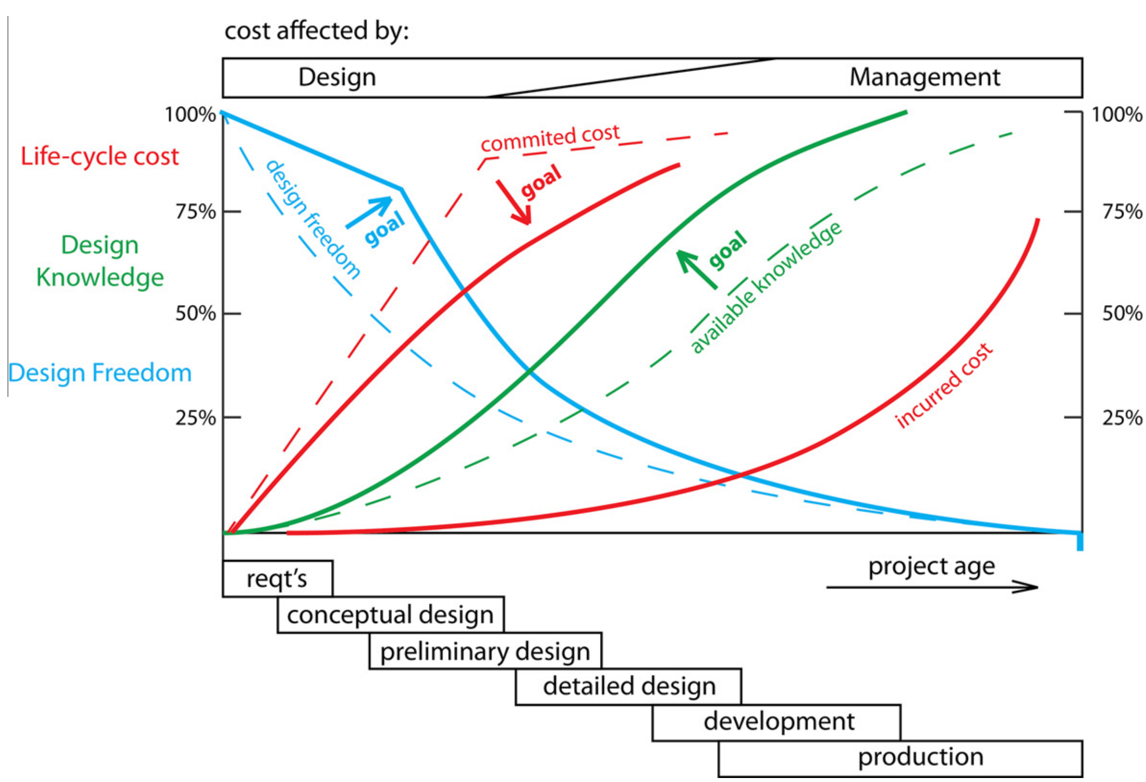

Fig. 2. Product life-cycle cost, design knowledge and freedom related to design process.

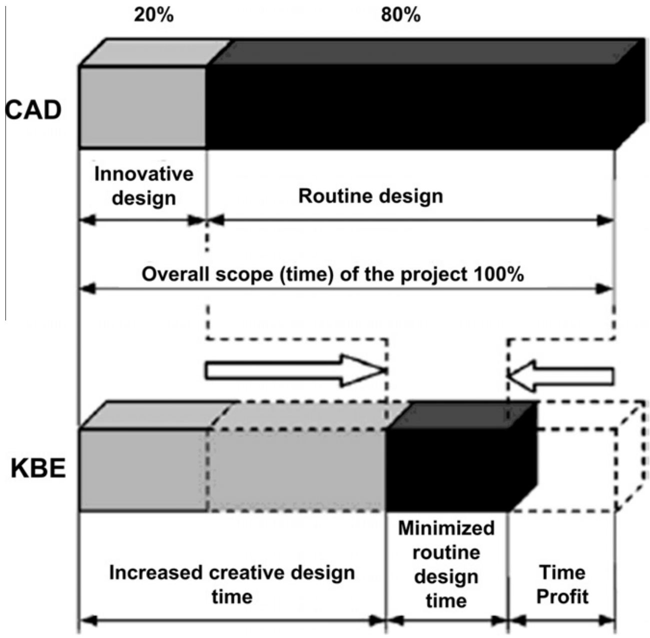

Fig. 3. Achievable design time allocation by KBE adoption [48].

manufacturing tools and processes [39,52]. To bring home the message that KBE can result in considerable time and cost savings while allowing for more creative work in conceptual and preliminary design, the results from a limited number of KBE projects are summarized in Table 2. Other quantified case studies are available in Cooper et al. [3].

It is important to note, however, that KBE is not by definition suitable for all design tasks. A number of authors note reasons for not adopting KBE [3,8,11,59]; the following as identified by Stokes [11] stand out:

- The design task is relatively straightforward and can be modelled and executed using less resources than a more demanding KBE approach.
- The organisation does not have the will, money or resources to introduce a KBE system. Nowadays, companies tend to move towards Commercial-Of-The-Shelft (COTS) solutions and tend to shy away from in-house software development, which is necessary in the case of KBE development.
- The design process consists of creative processes and products that are highly subject to change.
- The knowledge for the desired application is not available.
- The design process cannot be clearly defined; it is not possible to isolate and define particular stages in the design process.
- The technology in the design process is constantly changing.

As can be perceived from the list, these reasons are expressed qualitatively. In relation to this, there is no coherent framework to (quantitatively) assess whether a design task or process is suitable to be converted into a KBE representation. This point will be revisited in Section 4.

3.3. Methodological approaches to Knowledge-Based Engineering

A number of KBE methodologies are available to support the development of KBE applications and systems. By far the most well-known of these is the Methodology and software tools Oriented to Knowledge-Based Engineering Applications, or MOKA methodology. This methodology, based on eight KBE life-cycle steps and expressed in accompanying case-specific informal and formal models, is designed to take a project from inception towards industrialization and actual use [11]. The informal model consists of so-called ICARE forms, where the acronym stands for Illustrations, Constraints, Activities, Rules and Entities. These forms can be used to decompose and store knowledge elements. Subsequently, these elements can be linked to create a structured web of knowledge elements that together make up a representation of the problem domain to which users from multiple viewpoints can relate. When the problem knowledge has been converted into a structured representation, the next step is to formalize this knowledge in order to represent knowledge in a form that is acceptable to knowledge and software engineers and suitable for subsequent development of the KBE application. The formal model uses MML (Moka Modelling Language, an adaptation of UML) to classify and structure the ICARE informal model elements, which are translated into formal Product and Process models. The main elements of the MOKA methodology are illustrated in Fig. 3 – a more in-depth discussion of MOKA can be found in Stokes [11].

The main focus of MOKA lies with the 'Capture' and 'Formalize' steps of the KBE life-cycle (see Fig. 4). Herein lies a root cause for some of the missing ingredients of the MOKA methodology [9]:

- MOKA is focused on supporting the knowledge engineer (the person(s) responsible for capturing and formalizing the knowledge necessary for the KBE application) rather than the end user. The latter is however very important for both the development and the actual use of the KBE application, as the end user is typically the domain expert who holds the knowledge. Also, the end user should derive a clear benefit from using the KBE application while minimizing any extra workload, to improve acceptation, use and maintenance of the application.
- Knowledge representation mechanisms and supporting tools are not fully identified. This was done purposefully, as MOKA is intended as a neutral methodology. However, this choice has implications with respect to the accessability of knowledge in developed KBE applications (see Section 4.1).

Table 2
Selected KBE project results.

Reference	Subject	Effects
Van der Laan et al. [2]	Parametric modeling of movables for structural analysis	Up to 8% time savings in FE model generation (From 8 h to 1 h for specific instances)
Chapman & Pinfold [8]	Automotive body-in-white concept design	BIW mesh generation from 15 man weeks to 'minutes'
Choi et al. [16]	Composite aerospace structure: cost and weight estimation	Rapid evaluation of cost and weight for composite structures: supports trade-off capability
Kulon et al. [26]	Manufacturing process design: hot forging	New designs in hours rather than days or weeks. Supporting accessible knowledge base

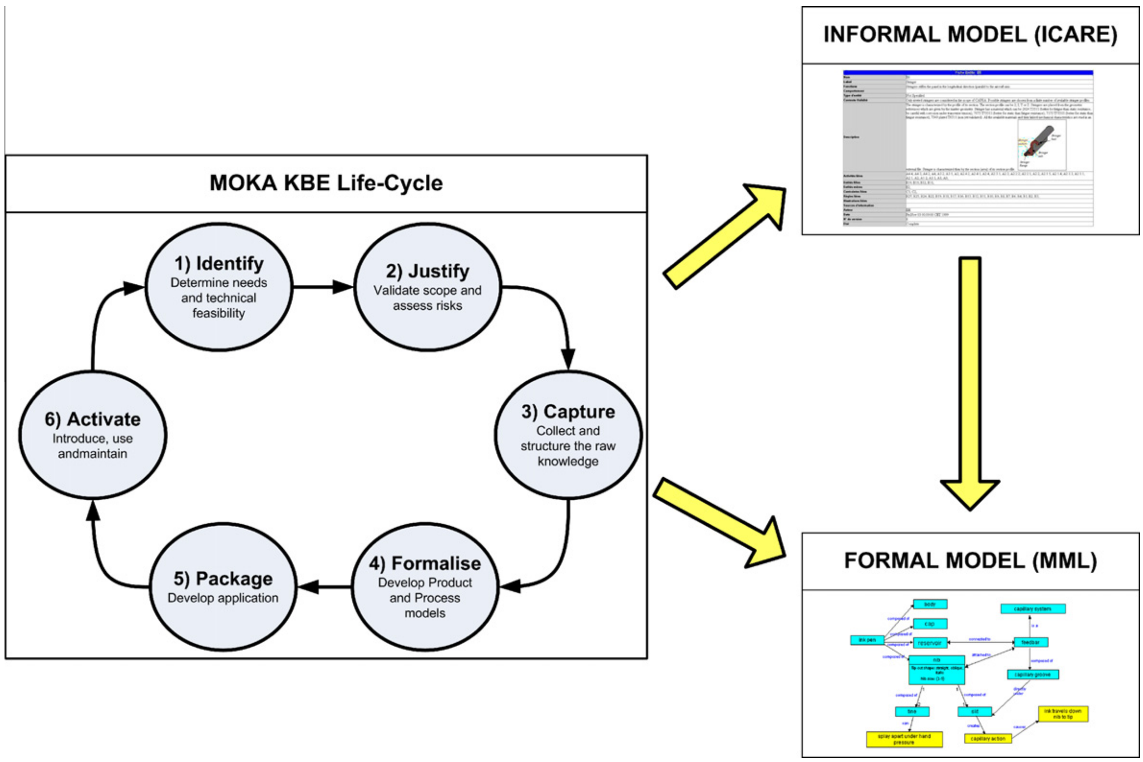

Fig. 4. MOKA methodology elements.

- MOKA is focused on the development of KBE applications. It does not give much attention to the use of the KBE applications in the design process itself and does not thoroughly consider maintenance and re-use of knowledge.
- MOKA is dominantly product-oriented, rather than process-oriented.

Another available KBE methodology is KOMPRESSA: Knowledge-Oriented Methodology for the Planning and Rapid Engineering of Small-Scale Applications [12]. This methodology aims to support KBE implementation at Small to Medium Enterprises (SMEs) and shares many principles with MOKA, with which it was developed in parallel. It distinguishes itself from MOKA by an increased emphasis on risk analysis and management, the increased use of activity diagrams to guide organizational and individual participants, and the assumption of little to no IT expertise on part of its users.

In response to the perceived shortcomings of MOKA, the KNOMAD methodology has been devised [9]. KNOMAD stands for Knowledge Nurture for Optimal Multidisciplinary Analysis and Design and is a methodology for the analytical utilization, development and evolution of multi-disciplinary engineering knowledge within the design and production realms. The KNOMAD acronym also represents KNOMAD's formalized process of: (K)nowledge capture; (N)ormalisation; (O)rganisation; (M)odeling; (A)nalysis; and (D)elivery. This methodology positions KBE within the design process, includes provisions for integration with knowledge representation capabilities to improve knowledge retention and application maintenance, and emphasises and addresses the role of the end user.

A final methodological approach towards KBE is the Design and Engineering Engine (DEE) advocated by La Rocca & Van Tooren [41] and Van der Laan & Van Tooren [53]. This approach incorporates KBE within an overall multidisciplinary design optimization (MDO) approach. A Multi-Model Generator (MMG) is used to describe a product family in terms of common building stones. The MMG has a generative capability to automatically instantiate geometry and an ability to create multiple perspectives on the product model. The MMG is connected to specific disciplinary analyses modules, which process information from the product model. The analysis results serve as inputs for the overall MDO process.

3.4. Identification of current shortcomings

Five major shortcomings of KBE have been distilled from the review sample.

(1) **Case-based, ad hoc development of KBE applications:** Development of KBE applications is still very much case based and happens on an ad hoc basis [10]: instead of using a structural framework or methodology to develop KBE applications, it seems that developers identify a problem and improvise a KBE solution based on a custom development process. This notion is confirmed by the review of KBE applications performed in this paper. It is most clearly visible by the wide-spread non-adherence to KBE design methodologies. From the 37 papers describing case studies, 81% (30 papers) did not explicitly adhere to a specific methodology. The practical impact of existing methodologies seems to be limited. The resulting case-based nature of KBE development is a significant problem. It can lead to knowledge loss due to poor modeling of the application and inadequacies in the used development language; it can cause knowledge misuse if the wrong kind of applications are developed; knowledge runs a danger of being under-utilized, due to an inability to share and re-use it at computer and human levels, and finally, maintenance costs will be higher due to non-standard development [10].

(2) **A tendency toward development of 'black-box' applications:** Another finding of the review is that current KBE development has a tendency towards 'black-box' applications – many applications (e.g. those in [28,36,39]) at best

represent captured knowledge as context-less data and formulas. There is no explication of formulas and the actual meaning and context of the captured knowledge, let alone provisions for capturing design intent. Part of it this is undoubtedly due to the difference between knowledge sources and knowledge engineers: one thinks in 'real-life' products and processes, the other thinks in models and programming. Previous research efforts have however shown the importance and validity of code annotation as a mechanism of linking data and formulas with the underlying knowledge [22]. In fact, KBE development is in real need of having a "round-trip" automatic feature that allows users and developers to go back from application code to the formal and informal models in order to validate, update and re-use the underlying knowledge. Such a feature has not yet been developed.

(3) **A lack of knowledge re-use:** The previous review findings tie in closely with the difficulty of re-using knowledge in KBE systems. Case-based black-box KBE applications do not particularly invite knowledge re-use. Aside from that, higher-level knowledge such as project constraint reasoning, problem resolution methods, solution generation strategies, design intent and supply chain knowledge [21] is often not captured, let alone re-used. Knowledge re-use is further complicated by the difficulty of sharing knowledge across (KBE) applications and platforms; as Bermell-Garcia [22] notes, 'using current data exchange standards, it is only possible to transfer an instance of the design (one state of the design), and not the knowledge embodied to generate it' – see also Fig. 4. An example of such a data exchange standard is STEP, which allows for exchange of geometry information, but cannot express any of the knowledge required for the creation of it. Clearly, new standards are required to enable knowledge sharing across applications and platforms.

(4) **A failure to include a quantitative assessment of KBE costs and benefits:** Another interesting research finding is that most KBE research fails to quantitatively illustrate the advantages and costs of KBE. 25 out of 37 case studies (67%) do not mention the resulting time or cost advantages associated with KBE adoption, let alone the more sensitive information about KBE development cost. An excellent example of how to perform and illustrate such a quantitative description of a KBE development effort is given in Embery et al. [32] – see Fig. 5. This figure compares two different product development cycles: one with traditional, manual design (the continuous blue line) where each design cycle leads to one generated design, versus the KBE design cycle where a KBE development must first be developed and is subsequently used in the design process. Fig. 5 illustrates that the KBE development time was equal to the time required for six traditional development cycles. The KBE development effort recoups this time quickly by having a much faster design iteration cycle than traditional design. A more systematic quantification effort has been performed by Corallo et al. [29], who use Activity Based Performance Measurement (ABPM) for cost-benefit assessment of KBE in new product development. Unfortunately, this quantification effort has been performed on a single case study, so validity, reliability and generalizability of the ABPM approach for KBE quantification are not known.

(5) **A lack of a (quantitative) framework to identify and justify KBE development:** A final KBE aspect that has not received much attention in literature is the assessment of KBE development opportunities. The MOKA handbook [11] presents some qualitative criteria for identification and justification of KBE opportunities and one paper by Emberey et al. [32] uses these and more criteria to assess whether a design task is suitable for KBE application development. However, despite these initiatives, no solid framework or method using both qualitative and quantitative aspects is available to determine whether a design task, product or process is suitable to develop a KBE application for.

In Section 4, research challenges are identified and discussed that have the potential to address these KBE shortcomings.

4. Research challenges in Knowledge-Based Engineering

In the preceding discussion, several KBE shortcomings have been identified. To remedy these shortcomings, the corresponding research challenges have to be addressed. These research challenges are discussed in detail below.

(1) **Improve methodological support for KBE:** First and most importantly, the discussed lack of exchange standards and excessive ad hoc character of KBE developments reflect methodological support for KBE as a key research challenge. To advance KBE research and development, it is advisable to move from case-based to methodology-guided projects. Obviously, methodology adherence cannot be enforced. Thus, the following critical question must be researched and answered: why are available KBE methodologies not used more often? Based on the answer to this question, existing methodologies may be subjected to improvement efforts or alternatives can be developed. A key issue for methodology adherence may be the availability of supporting tools and technologies for KBE implementation; currently, a "technology gap" exists, with a lack of tools dedicated to support MOKA's informal and formal models, with the notable exception of PC-PACK [61]. Also, automatic conversion of formal models into application code by translators is an underdeveloped area of research. As has been noted before, there is also no "round-trip" feature for maintenance and re-use of knowledge and associated application code. As the literature sample did not give sufficient insight into the technological aspect of KBE, this issue is not further explored here: it will be explored in more detail in future research.

(2) **Moving beyond black-box KBE applications: Two** specific research issues are observed that must be addressed to move towards accessible, open KBE applications.

   (a) **Transparency of KBE applications:** The necessity for increased transparency in KBE systems is a well-noted research issue [5,15,49]. In order to move away from black-box applications towards applications with more user-friendly, flexible and adaptable knowledge bases, more attention needs to be paid to KBE knowledge base component development. First of all, means must be provided to substantiate knowledge used in KBE: the underlying knowledge and supporting documentation for the KBE application should be directly accessible. Direct interfacing with knowledge management applications [33,62] or PLM solutions [8,15,16] can be used to achieve this. Also, it should be visible what the application is actually doing. Some research has been ongoing to increase visibility of knowledge in KBE applications [22,51,60]. Another important aspect to achieve transparency is management of the evolution of applications; this is part of the previously identified "round-trip" functionality that should be developed. More advanced knowledge management techniques and features such as knowledge life-cycle management may be used as part of this development.

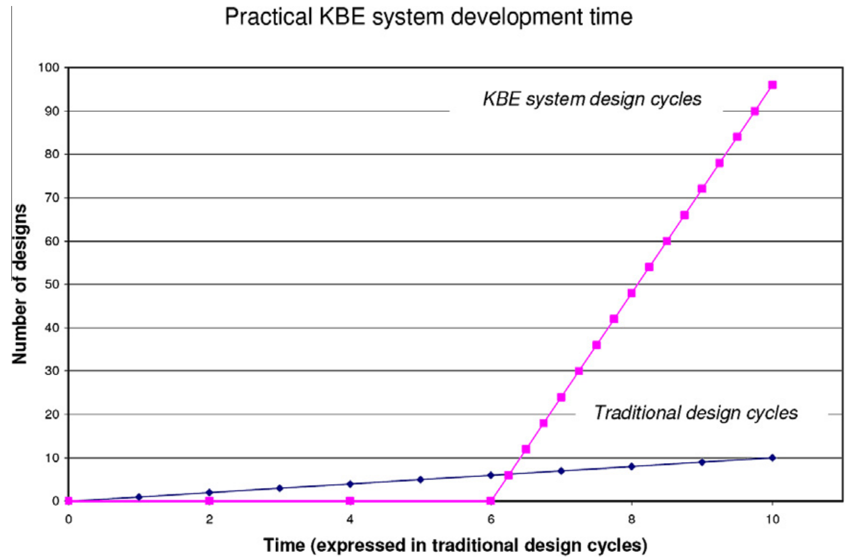

Fig. 5. KBE system development time versus traditional design time [32].

(b) **Enriching the semantics of knowledge models and their traceability in KBE implementation:** An interoperability research agenda addresses only part of the wider KBE issue of knowledge capitalization through effective re-use, sharing and maintenance. Achieving interoperability enables the "data connection" between heterogeneous systems to be used into KBE tools and vice versa. Further exploitation of data entities in KBE automation tools requires a "knowledge connection" where not only the syntactic content of the data but its common meaning is exchanged, an essential principle of the so-called "Semantic Web" [63].

This is opposed to traditional KBE practice in which information models (typically object-oriented) are contained within the applications. Although this approach has proven itself for fast development of ad hoc KBE tools, it has also limited the longer-term potential to re-use, share and maintain the underlying knowledge models. Historically, the previously mentioned MOKA project picked up on the issue as its business proposition for better management of the knowledge in KBE applications. The project was leveraged by the state of the art on Knowledge Engineering (KE) compiled in the CommonKADS methodology [57], resulting in the decoupling of engineering knowledge (informal models) from implementation software models and code (formal models).

A parallel can be drawn on how KE has lead to the Semantic Web and similarly on how "post-MOKA" KBE can take a similar step. In setting up the analogy as a research challenge, a direction for future KBE technology can be perceived. As in the Semantic Web vision, KBE technology shall rely on semantically-rich knowledge resources distributed across the web or the enterprise networks. The ability of computer systems to interpret these models can be exploited by KBE architects to assemble tools that generate trustable data addressing the "ad hoc" engineering tasks at hand. The achievement of this scenario in KBE will require attention to the following aspects:

- Supporting the creation of tool-specific knowledge models working on the assumption that not all the knowledge resources to suit "ad hoc" KBE applications are available. A key principle to maintain is the decoupling of knowledge models from code while maintaining a level of traceability between both.
- Supporting the re-use of existing domain knowledge models into single applications rather than attempting to re-build these models in an ad hoc manner every time KBE tools are created.
- Supporting the assembly of applications whose results are based on connecting to diverse software tools rather than relying only on the CAD tool to which the KBE languages are tightly integrated.

A possible roadmap to address these challenges is to embrace the concept of Model Driven Engineering (MDE) in a similar way to how CommonKADS was adopted by MOKA. MDE is a generic term to describe an emerging way of creating software systems that allows its business semantics to be separated from code implementations [64]. MDE is not a pure software engineering best practice. Its roots can be traced back to the KE field where the previously mentioned separation between knowledge and implementation of software was advocated. Not only software engineering but also system engineering practitioners have exploited this concept to link their software application codes to domain models and even to support the automatic generation of programs. A key factor on the adoption of MDE is the progressive emergence of semantically-rich models known as Domain Specific Modeling (DSM). These modelling tools enable to describe domain knowledge not only in a structured manner but also in a way that is readable by software systems [65].

(3) **Effectively sourcing and re-using knowledge:** In KBE development, as in any software intensive activity, the effective sourcing of data, information and knowledge has important implications with respect to the quality of the resulting applications. Early KBE industrial tools like the ones mentioned in Cooper et al. [3] had a strong focus on "integrated modeling" as a mechanism to bring together rules from different engineering domains. In these early applications a large amount of the knowledge was captured directly from domain experts. Simultaneously, in the last decades a culture of "data curation" has emerged in engineering organizations to cope with the long lifecycles of information which often outlive the products themselves [66]. A benefit of this curation is the ability to reduce the duplication of data across systems by introducing the concept of "mastering data". Research is needed to better and more explicitly

exploit the potential of mastering data in the KBE field in order to retrieve and re-use data, information and knowledge available in diverse repositories. With respect to knowledge re-use, an important issue to address is the creation of standards to facilitate the interoperability of KBE platforms and applications through the sharing of knowledge (models). As has been noted in Section 4.1, there currently are no standards for this issue, though a call for projects was announced for this topic in 2008 [15].

(4) **KBE success metrics:** Despite efforts by Emberey et al. [32] and Corallo et al. [29] as noted in Section 4.1, the review of the literature sample has shown that KBE success metrics on KBE development and implementation projects are most often not reported. Primarily, cost and benefit figures for KBE development are not reported; it is difficult to judge the business proposition for KBE. Intellectual Property (IP) considerations and funding concerns may prohibit quantification of KBE efforts in research literature. Also, selection bias may prevent less successful KBE initiatives from reaching publication. From a performance metrics point of view, potential developments with respect to the distribution of KBE applications throughout the enterprise and supply chain network open up a promising scenario where it will be possible to quantify and assess the use of tools (e.g. number of runs). Such a scenario is only viable when explicit workflow automation is applied. In this way, performance metrics can be recorded (e.g. number of runs, processing times, storage of inputs and outputs). The storage of such metrics opens up the way towards quantification of KBE use and performance.

(5) **Assessment framework for KBE opportunities:** Currently, no solid framework or method is available to determine whether a design task, product or process is suitable to develop a KBE application for. For instance, when is something a routine process and suitable for KBE? How can one convincingly distinguish between routine and non-routine processes? Such considerations are seldomly reported in KBE literature, a notable exception being Emberey et al. [32] which uses a qualitative assessment framework for a single use case. KBE is in need of a coherent framework to assess whether a design task or process is suitable to be converted into a KBE representation. Preferably, this framework would include quantifiable metrics next to qualitative considerations. An example of an important qualitative consideration is team composition: what set of skills and which associated team composition is necessary to deliver the required knowledge and implement the required KBE application? Finally, to convincingly make the business case for (or against) KBE, supporting methods for quantification through cost-benefit analyses should be available.

A final element of KBE has not yet been discussed, as it is outside the realm of methodological support for KBE. This element is the potential to pursue distributed deployment and use of KBE tools. This opportunity is derived from the realization that KBE tools can be shared not only locally (e.g. in specific design teams), but also across the enterprise and across the supply chain. Distributed deployment and use of KBE tools, for instance through web-based applications, opens up the possibility for collaborative capture and use of knowledge as well as performing collaborative design. The possibility of relying on shared knowledge and specifications is an attractive proposition for extended enterprises, though IP concerns must be addressed.

5. Conclusions

After more than three decades of existence, the conceptual foundations of KBE remain hard to define, particularly as an exhaustive set of technologies such as CAD, AI and others. This paper instead presents KBE as a "way of working" intended to deliver engineering design automation in scenarios where the retention of knowledge is critical. A number of critical research issues have been identified. These need to be addressed before the promise of KBE can be more convincingly achieved.

A key research challenge is formed by the need for better support by and adherence to KBE methodologies in KBE development projects. To achieve this, a number of issues should be researched, the foremost of which is closing the "technology gap": the lack of tools and technologies to support cost-effective KBE development. This issue will be subject for further research. Other important research issues include: achieving transparency of KBE systems, enabling knowledge sourcing and re-use through data mastering and standard development, enriching the semantics of knowledge models and ensuring traceability for implemented KBE solutions, reporting KBE success metrics, and providing an assessment framework for KBE opportunities.

KBE holds particular promise as an enabling technology for mass customization through the opportunity of distributed deployment of KBE tools across the extended enterprise. Knowledge sourcing, re-use and traceability are important aspects for performing shared design between OEM and supply chain partners, while the automation aspect of KBE has the potential to significantly reduce design times. This brings the potential to respond quickly to customer requirements and generate a range of variant designs to meet specific requirements: thus, KBE is an enabling technology for mass customization.

The field of Knowledge-Based Engineering remains to hold significant promise, but has significant challenges to investigate and address for the coming decade.

Acknowledgements

We would like to thank Adrian Murton (Airbus UK) as well as the reviewers for their helpful comments and suggestions to improve this paper.

References

[1] M.M. Tseng, J. Jiao, Design for mass customization, Annals of the CIRP 45 (1996) 153–156.
[2] N. Wognum, A. Trappey, PLM challenges, Advanced Engineering Informatics 22 (2008) 419–420.
[3] S. Cooper, I.S. Fan, G. Li, Achieving Competitive Advantage Through Knowledge Based Engineering – A Best Practice Guide, Cranfield University, Bedford, UK, 2001.
[4] F. Elgh, Supporting management and maintenance of manufacturing knowledge in design automation systems, Advanced Engineering Informatics 22 (2008) 445–456.
[5] F. Elgh, M. Cederfeldt, Documentation and Management of Product Knowledge in a System for Automated Variant Design: A Case Study, in: J. Pokojski, S. Fukuda, J. Salwiński (Eds.), New World Situation: New Directions in Concurrent Engineering, Springer, London, 2010, pp. 237–245.
[6] D. Cooper, G. La Rocca, Knowledge-based Techniques for Developing Engineering Applications in the 21st Century, in: 7th AIAA ATIO Conference, AIAA, Belfast, Northern Ireland, 2007.
[7] S. Danjou, N. Lupa, P. Koehler, Approach for automated product modeling using knowledge-based design features, Computer-Aided Design and Applications 5 (2008) 622–629.
[8] P. Bermell-Garcia, I.S. Fan, Practitioner requirements for integrated knowledge-based engineering in product lifecycle management, International Journal of Product Lifecycle Management 3 (2008) 3–20.
[9] R. Curran, W.J.C. Verhagen, M.J.L Van Tooren, T.H. van der Laan, A multidisciplinary implementation methodology for knowledge based

engineering: KNOMAD, Expert Systems with Applications 37 (2010) 7336–7350.
[10] P. Sainter, K. Oldham, A. Larkin, Achieving benefits from Knowledge-Based Engineering systems in the longer term as well as in the short term, in: Proceedings International Conference on Concurrent Enterprising, Toulouse, France, 2000.
[11] M. Stokes, Managing Engineering Knowledge–MOKA: Methodology for Knowledge Based Engineering Applications, Professional Engineering Publishing Limited, London, 2001.
[12] P.J. Lovett, A. Ingram, C.N. Bancroft, Knowledge-based engineering for SMEs – a methodology, Journal of Materials Processing Technology 107 (2000) 384–389.
[13] J. Pokojski, K. Szustakiewicz, M. Gil, Engineering Knowledge Modeling in Design, in: J. Pokojski, S. Fukuda, J. Salwiński (Eds.), New World Situation: New Directions in Concurrent Engineering, Springer, London, 2010, pp. 287–297.
[14] O. Kuhn, Methodology for Knowledge-Based Engineering Update, in: École Doctorale Informatique et Mathématiques Lyon, L'Université Claude Bernard Lyon, Lyon, 2010. pp. 202.
[15] I.S. Fan, P. Bermell-Garcia, International standard development for knowledge based engineering services for product lifecycle management, Concurrent Engineering-Research and Applications 16 (2008) 271–277.
[16] Q.C. Ma, X.W. Liu, Review of knowledge based engineering with PLM, Applied Mechanics and Materials 10–12 (2007) 127–131.
[17] Z.N. Guo, J. Sheffield, A paradigmatic and methodological examination of knowledge management research: 2000–2004, Decision Support Systems 44 (2008) 673–688.
[18] Y. Rezgui, C.J. Hopfe, C. Vorakulpipat, Generations of knowledge management in the architecture, Engineering and Construction Industry: An evolutionary perspective, Advanced Engineering Informatics 24 (2010) 219–228.
[19] H.A. Simon, The Sciences of the Artificial, 3rd ed., MIT Press, Cambridge, MA, 1996.
[20] S. Ammar-Khodja, N. Perry, A. Bernard, Processing knowledge to support knowledge-based engineering systems specification, Concurrent Engineering-Research and Applications 16 (2008) 89–101.
[21] D. Baxter, J. Gao, K. Case, J. Harding, B. Young, S. Cochrane, S. Dani, An engineering design knowledge reuse methodology using process modelling, Research in Engineering Design 18 (2007) 37–48.
[22] P. Bermell-Garcia, A metamodel to annotate knowledge based engineering codes as enterprise knowledge resources, in: School of Applied Sciences, Cranfield University, Cranfield, 2007. pp. 150.
[23] P. Bermell-Garcia, I.S. Fan, A KBE System for the Design of Wind Tunnel Models Using Reusable Knowledge Components, in: International Congress on Project Engineering, Barcelona, 2002.
[24] C.B. Chapman, M. Pinfold, Design engineering - a need to rethink the solution using knowledge based engineering, Knowledge-Based Systems 12 (1999) 257–267.
[25] C.B. Chapman, M. Pinfold, The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure, Advances in Engineering Software 32 (2001) 903–912.
[26] J.W. Choi, Architecture of a knowledge based engineering system for weight and cost estimation for a composite airplane structures, Expert Systems with Applications 36 (2009) 10828–10836.
[27] J.W. Choi, D. Kelly, J. Raju, A knowledge-based engineering tool to estimate cost and weight of composite aerospace structures at the conceptual stage of the design process, Aircraft Engineering and Aerospace Technology 79 (2007) 459–468.
[28] J.W. Choi, D. Kelly, J. Raju, C. Reidsema, Knowledge-based engineering system to estimate manufacturing cost for composite structures, Journal of Aircraft 42 (2005) 1396–1402.
[29] A. Corallo, R. Laubacher, A. Margherita, G. Turrisi, Enhancing product development through knowledge-based engineering (KBE): a case study in the aerospace industry, Journal of Manufacturing Technology Management 20 (2009) 1070–1083.
[30] F. Elgh, Modelling and management of manufacturing requirements in design automation systems, complex systems concurrent engineering: collaboration, Technology Innovation and Sustainability (2007) 321–328.
[31] F. Elgh, Supporting Management and Analysis of Quotations in a Design Automation Approach to Customization, in: J. Pokojski, S. Fukuda, J. Salwiński (Eds.), New World Situation: New Directions in Concurrent Engineering, Springer, London, 2010, pp. 401–409.
[32] C.L. Emberey, N. Milton, J.P.T.J. Berends, M.J.L. Van Tooren, S.W.G. Van der Elst, B. Vermeulen, Application of Knowledge Engineering Methodologies to Support Engineering Design Application Development in Aerospace, in: 7th AIAA Aviation Technology, Integration and Operations Conference (ATIO), Belfast, Northern Ireland, 2007.
[33] I.S. Fan, G. Li, M. Lagos-Hernandez, P. Bermell-Garcia, M. Twelves, A Rule Level Knowledge Management System for Knowledge Based Engineering Applications, in: ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2002), ASME, Montreal, Quebec, Canada 2002, pp. 813–821.
[34] A. Kamrani, A. Vijayan, A methodology for integrated product development using design and manufacturing templates, Journal of Manufacturing Technology Management 17 (2006) 656–672.
[35] R. Klein, Towards an Integration of Engineering Knowledge Management and Knowledge Based Engineering, in: M. Spitalni (Ed.), Proceedings of the CIRP Conference on Advanced Engineering Information Systems, Haifa, Israel, 2000.
[36] K.H. Ko, K. Pochiraju, S. Manoochehri, An embedded system for knowledge-based cost evaluation of molded parts, Knowledge-Based Systems 20 (2007) 291–299.
[37] O. Kuhn, P. Ghodous, T. Dusch, P. Collet, Collaboration for Knowledge-Based Engineering Templates Update, in: J. Pokojski, S. Fukuda, J. Salwiński (Eds.), New World Situation: New Directions in Concurrent Engineering, Springer, London, 2010, pp. 277–285.
[38] J. Kulon, P. Broomhead, D.J. Mynors, Applying knowledge-based engineering to traditional manufacturing design, International Journal of Advanced Manufacturing Technology 30 (2006) 945–951.
[39] J. Kulon, D.J. Mynors, P. Broomhead, A knowledge-based engineering design tool for metal forging, Journal of Materials Processing Technology 177 (2006) 331–335.
[40] S. Kundu, Application of Design Automation to Reduce Cycle Time of Hydro Turbine Design, in: Industrial Application, 2008, pp. 113–117.
[41] G. La Rocca, M.J.L. Van Tooren, Enabling distributed multi-disciplinary design of complex products: a knowledge based engineering approach, Journal of Design Research 5 (2007) 333–352.
[42] I. Mendikoa, M. Sorli, J. Barbero, A. Carrillo, Distributed Product Design and Manufacturing Based on KBE, in: W. Shen, K.-M. Chao, Z. Lin, J.-P. Barthès, A. James (Eds.), Computer Supported Cooperative Work in Design II, Springer, Berlin/Heidelberg, 2006, pp. 404–413.
[43] M. Ruschitzka, A. Suchodolski, J. Wróbel, Ontology-Based Approach in Hybrid Engineering Knowledge Representation for Stamping Die Design, in: J. Pokojski, S. Fukuda, J. Salwiński (Eds.), New World Situation: New Directions in Concurrent Engineering, Springer, London, 2010, pp. 227–235.
[44] M. Sandberg, Knowledge Based Engineering - In Product Development, in: Technical Report, Lulea University of Technology, Department of Applied Physics and Mechanical Engineering, Division of Computer Aided Design, Lulea, 2003.
[45] A. Schneegans, F. Ehlermann, Applying KBE Technologies to the Early Phases of Multidisciplinary Product Design, in: F.L. Krause (Ed.), 17th CIRP Design Conference, Berlin, Germany, 2007, pp. 587–595.
[46] E. Shehab, H. Abdalla, An intelligent knowledge-based system for product cost modelling, The International Journal of Advanced Manufacturing Technology 19 (2002) 49–65.
[47] E. Shehab, H. Abdalla, A cost-effective knowledge-based reasoning system for design for automation, Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture 220 (2006) 729–743.
[48] W. Skarka, Application of MOKA methodology in generative model creation using CATIA, Engineering Applications of Artificial Intelligence 20 (2007) 677–690.
[49] S. Sunnersjo, M. Cederfeldt, F. Elgh, I. Rask, A transparent design system for iterative product development, Journal of Computing and Information Science in Engineering 6 (2006) 300–307.
[50] S.W.G. Van, M.J.L. der Elst, Van. Tooren, Application of a Knowledge Engineering Process to Support Engineering Design Application Development, in: R. Curran, S.-Y. Chou, A. Trappey (Eds.), Collaborative Product and Service Life Cycle Management for a Sustainable World, Springer, London, 2008, pp. 417–431.
[51] A.H. Van der Laan, Knowledge based engineering support for aircraft component design, in: Design of Aircraft and Rotorcraft, Faculty of Aerospace Engineering, Delft University of Technology, Delft, 2008, pp. 254.
[52] A.H. van der Laan, R. Curran, M.J.L. van Tooren, C. Ritchie, Integration of friction stir welding into a multidisciplinary aerospace design framework, Aeronautical Journal 110 (2006) 759–766.
[53] A.H. van der Laan, M.J.L. Van Tooren, Parametric modeling of movables for structural analysis, Journal of Aircraft 42 (2005) 1606–1614.
[54] B. Vermeulen, Knowledge based method for solving complexity in design problems, in: Design of Aircraft and Rotorcraft, Faculty of Aerospace Engineering, Delft University of Technology, Delft, 2007. pp. 126.
[55] J. Zweber, M. Blair, H. Kamhawi, G. Bharatram, A. Hartong, Structural and manufacturing analysis of a wing using the adaptive modelling language, in: American Institute of Aeronautics and Astronautics, AIAA-98-1758, 1998.
[56] M. Alavi, D.E. Leidner, Review: knowledge management and knowledge management systems: conceptual foundations and research issues, Mis Quarterly 25 (2001) 107–136.
[57] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga, Knowledge Engineering and management: the CommonKADS methodology, MIT Press, Cambridge, MA, 1999.
[58] W.J. Fabrycky, B.S. Blanchard, Life-Cycle Cost and Economic Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1991.
[59] D.S. Darai, S. Singh, S. Diswas, Knowledge Engineering - an overview, International Journal of Computer Science and Information Technologies 1 (2010) 230–234.
[60] P. Bermell-Garcia, I.S. Fan, A. Murton, Towards the semantic interoperability between KBE and PLM systems, in: Proceedings of International Conference on Engineering Design (ICED07), The Design Society, Paris, France, 2007.
[61] Epistemics, PCPACK, in.
[62] W.J.C. Verhagen, P. Bermell-Garcia, P. Mariot, D. Ruiz, J. Cotton, R. Redon, R. Curran, Knowledge-Based Cost Modelling of Composite Wing Structures, International Journal of Computer Integrated Manufacturing, (In Press).
[63] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intelligent Systems and Their Applications 16 (2001) 72–79.

[64] D.C. Schmidt, Model-driven engineering, Computer 39 (2006) 25–31.
[65] S. Kelly, J.P. Tolvanen, Domain-Specific Modeling – Enabling Full Code Generation, John Wiley & Sons, Hoboken, New Jersey, 2008.
[66] C. McMahon, M. Giess, S. Culley, Information management for through life product support: The curation of digital engineering data, International Journal of Product Lifecycle Management 1 (2005) 26–42.