

A new approach of a global knowledge-based engineering infrastructure

Markus Salchner^a , Mario Hirz^a , Severin Stadler^b  and Jonathan Ameye^b

^aGraz University of Technology, Austria; ^bMAGNA STEYR Engineering AG & Co KG, Austria

ABSTRACT

The ongoing globalization in almost all markets leads to a wide range of IT-related challenges. Focusing on development processes in mechanical engineering disciplines as they occur in automotive industry, globalization requires a continuously and location-independent provision of software tools and project-oriented data. This involves both supply of a large amount of different commercial software packages for various engineering disciplines as well as handling of large amount of data, configurations, different projects, sites and employee-related information. The broad field of computational design and engineering disciplines within automotive development requires a strong interaction in view of efficient software applications. Especially the CAD-based design phase is characterized by a cooperation of manufacturer and supplier, which often use dissimilar CAD software with different versions or even different vendors. Previous research work focused on development and application of knowledge-based engineering (KBE) applications within a multiple-CAD environment. A method has been proposed for efficient development of KBAs (Knowledge Based Applications), which are able to concurrently support multiple CAD systems and releases. Based on this fundamental approach, the next logical step includes the development of an efficient and secure distribution environment for KBAs. In this context, the present publication introduces a new approach for management and distribution of KBAs, project-related CAD environments and templates, as well as product data quality (PDQ) methods in a worldwide context.

KEYWORDS

Knowledge-based engineering; design automation; multi-CAD; globalization

1. Introduction

The broad field of computational design and engineering disciplines within automotive development requires a strong interaction in view of efficient software applications. Besides the automotive sector, many industries and companies are faced with this challenge. Furthermore, a collaboration between manufacturer and supplier, and also between different departments complicate an efficient interaction. In this context, the umbrella term KBE (Knowledge Based Engineering) is often used. This technique enables the improvement of product quality while reducing the development effort and time. A literature review of KBE definition, research challenges, future trends and technologies mainly leads to the works of Verhagen [29], Liao [16] and Reddy [21]. The work from Reddy, published in 2015, points out an increasing interest on this research topic. Due to the broad field of KBE, this introduction is used to clarify the topic and to clearly define the field of research in this paper. In literature, KBE is often cited as knowledge-based methods, which are used to capture and reuse engineering knowledge. In this context, the slightly more detailed

notion from Stokes “use of advanced software technologies to capture and reuse product and process knowledge in an integrated way” is also often cited [25]. La Rocca [15] introduced three knowledge sections – Knowledge Management (KM), Knowledge Engineering (KE) and Knowledge Based Engineering (KBE). KE deals with the problem identification, the capturing and formalization of knowledge. In this work the KBE term is exclusively related to the development of KBE applications. This comprises for example the integration and interaction with and within the CAD system. The KM section relates to the deployment and integration of the developed KBE applications. Furthermore, this section is involved in the problem identification task.

In our opinion, and in the present paper, the umbrella term KBE is used to define the *development*, *deployment* and *management* of tailor-made software applications, which reduce engineering effort and simultaneously improve efficiency and quality of development processes and subsequently of the product. In the present work, these software applications - KBAs (Knowledge based applications) - include standalone executable files

(* .exe), which use the application programming interfaces (APIs) of CAD systems and further third party applications. For the efficient development of these KBAs a novel KBE framework has been developed, introduced in a previous work [22]. This framework comprises the definition of global directives, programming guidelines, a multi-CAD approach and the superordinate infrastructure.

The present paper focuses on the secure distribution and efficient deployment of the developed KBAs as well as the associated customization and management tasks in a worldwide context. Considering the KBE lifecycle loop, which consists of the five main parts creation, capturing, formalizing, implementation and activation of knowledge [25], this paper relates to the activation section. Besides the deployment of KBAs, the present work comprises the project-related configuration and customization of the KBAs as well as the required modifications within the CAD system. Depending on the related industry field, the CAD settings can strongly differ from project to project; especially the supplier industry is faced with this varying environment.

The present research work is structured in three main Sections. The introduction covers on the one hand different state of the art software deployment methods. A literature survey as well as the underlying scientific research questions can be found in this Section. Section two comprises the proposed approach including the main components and required predefined/-developed environments. The final Section three illustrates and describes the validation of the approach in real testing environment.

To sum it up, the presented approach provides a method for the secure, efficient and generic deployment of KBAs. Supplementary, the project-related CAD settings are managed in a centralized database and the client environment is configured on demand. Therefore, the proposed approach fulfills four main objectives:

- Management of various engineering projects
- Global deployment, configuration and management of KBAs
- Project related configuration of the CAD system – on demand
- Feedback systems

1.1. Use of existing (commercial) deployment solutions

Focusing on development processes in mechanical engineering disciplines, as they occur in automotive industry, globalization plays a prominent role and requires continuously and location-independent provision of software

tools and project-oriented data. This involves both, supply of a large amount of different commercial software packages for various engineering disciplines, as well as handling of large amount of data, configurations, different projects, sites and employee-related information. The installation and configuration of commercial software packages like Microsoft Windows, Office [18] or basic installations of CAD systems is not the focus of this paper, as well as the setup and deployment of product data management (PDM) systems like Teamcenter [24] or Enovia [3].

As already mentioned in the previous Section, the introduced approach is focused on the deployment of KBAs and not on the distribution of commercial software packages. Exemplary commercial software that can be used for the deployment of commercial software packages are Matrix24 [17] and Citrix [1]. The client management system, named OPSI [20], is an example for an open-source package for the deployment of commercial software packages. This tool can be used for software and operation system deployment as well for monitoring and patch management, but it is not suitable for KBAs. Due to the limited customization options, license costs and installation effort, all mentioned above software is not suitable for the distribution of KBAs and the configuration of CAD systems. Furthermore, these solutions do not support the management of various engineering projects.

1.2. Literature survey

The literature survey for knowledge based system KBS/KBE/KM approaches and its future trends resulted in three main works:

- “Knowledge management technologies and applications-literature review from 1995 to 2002”, [16]
- “A critical review of Knowledge-Based Engineering: An identification of research challenges”, [29]
- “Knowledge Based Engineering: Notion, Approaches and Future Trends”, [21]

The main outcome of the literature review is that there are plenty research challenges in this broad engineering discipline. The time and thus cost intensive building and implementation of a KBS/KBE/KM environment is often mentioned as main drawback. Furthermore, the development trend toward black-box systems and the lack of knowledge reuse are cited as major shortcomings in current KBS/KBE/KM development. The complete product development process can often not be optimized due to localized implementation of the KBE system. Many research works use KBE methods, which are integrated

in CAD systems, whereby reuse and distribution are limited automatically. Web-based approaches are often cited, when changing customers have to be considered. Present KBE approaches have a lack regarding quantitative cost and benefit assessment of the applications, as stated in [29].

Exemplary research challenges in present KBS/KBE/KM development are improving methodological support, developing transparent applications, effectively sourcing and reusing of knowledge and also the field of artificial intelligence is from research interest. Another interesting scientific field includes the deployment of KBE applications across enterprises, across the supply chain and enhancement of collaborative possibilities and potentials. The proposed approach in the present work is focused on this research gap and presents a global infrastructure for deployment of KBAs. [16], [21], [29]

1.3. Software deployment approaches – state of the art

This Section gives an overview of state of art software deployment approaches. Before discussing the deployment itself, the content, what should be deployed, has to be clarified. Is there one single file or are there multiple files of the software to be distributed? Are there any relations between files? Each .NET application consists of one or more files (*.exe, *.dll, *.config . . .). Common applications consist of one executable file and some *.dlls

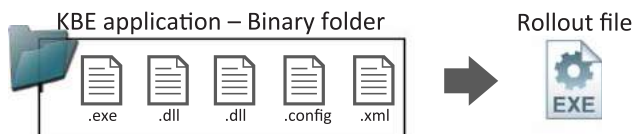


Figure 1. “All in one” executable.

(dynamic link libraries) that are basically class libraries. The executable file invokes these libraries and performs the coded automation routine. This requires by default that all files have to be in the same directory folder. Software updates can then be carried out by the replacement of the specific file/s.

The entry point of the approach is that only one single application file is distributed - a so-called “All in one” executable (Figure 1). This minimizes the amount of different files, which have to be handled by a distribution portal. Considering a midsize KBA that consists of about five to ten various programming libraries, the complexity can be reduced dramatically. One obvious drawback of this concept is, that software updates cannot be done by replacing single libraries. Instead, the entire application has to be released. This disadvantage may become significant within large software packages like CAD systems or operating systems, where an installation is mandatory. Company-specific and internal KBAs normally do not require an installation instance and they have a manageable size – some megabytes. Therefore, this seeming drawback isn’t one for the deployment of KBAs. To sum up, in the present approach there is only one single application file (*.exe), which has to be released for the client machines.

A suitable software distribution approach (Figure 2) for small and medium-sized enterprises (SMEs) within straightforward project environment and small customer-related influences is the usage of a company wide network drive. Therefore, the design engineer can execute the required KBAs from a centralized network storage. The responsible KBE administrator can update and maintain the applications with minimum effort. Using this approach, increased effort has to be considered, like management of user or project related access rights or even licensing models. Furthermore, the KBE administrator

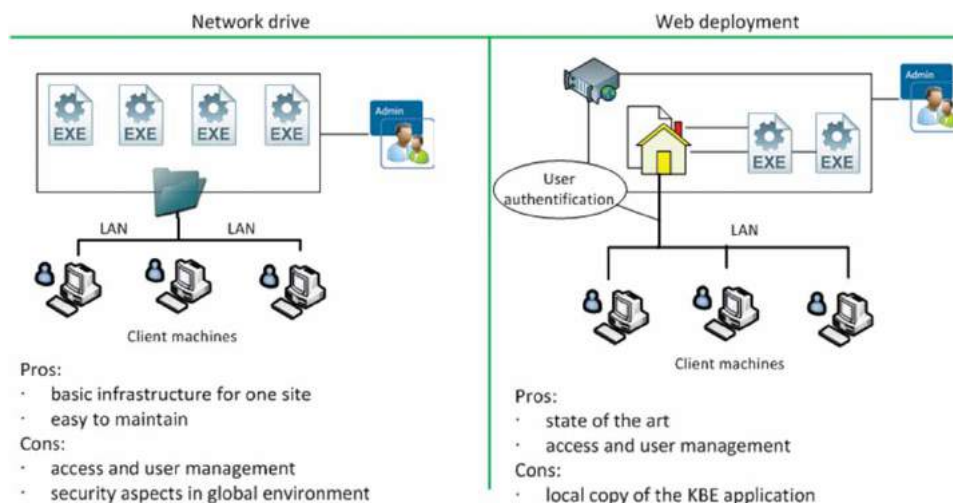


Figure 2. Common software distribution approaches.

cannot guaranty that all design engineers use the latest software version or they do not make an illegal copy of the software. The drawback of the update behavior can be avoided by using the integrated publication function of the Visual Studio environment. It allows a very simple application deployment selecting the installation medium (folder, web site or CD) and the possibility to define an update behavior. Therefore, the developer has to define the path to stored updates (folder or web site) and the update behavior (after or before application start, update interval in days). Considering multi-site company structures, secure web access to the network drive, like virtual private network (VPN) [19] connection has to be set up. Figure 2, left, illustrates this approach of a local area network (LAN). [18] [19] [26]

Another possibility to distribute KBAs is based on a download portal, as it is used for different commercial applications. The KBE administrator can provide the applications and updates on a web server, where registered users can download the required applications. Depending on the web server configuration, the download portal page can be accessed from everyone via the internet or just from computers within the LAN - as illustrated in Figure 2, right. This approach is quite similar to the previous one, but it includes the advantage, that user access can be managed in a proper way by assigning roles or other project-related qualifiers. However, users receive a copy of the KBAs and the KBE administrator has no influence on the further use or misuse of these applications.

With these two common approaches, it is not possible to control the client access rights, to fulfill security aspects, to get reasonable feedbacks and to enable an efficient global distribution of KBAs. Furthermore, project-related requirements e.g. from CAD system- or PDQ-settings cannot be implemented.

1.4. Project-related CAD and KBAs configuration

The presented global deployment approach has to comprise two project-related components:

- Configuration of the KBAs – different settings
- Configuration of the CAD system – on demand on each client computer

Each project within a development process has a predefined environment, including guidelines, rules, naming conventions, etc. A holistic KBA has to meet these requirements using an object oriented, agile, polymorphic programming approach, and the distribution environment has to be able to support this layout. A simple example includes the usage of different KBA

configuration files. This type of files defines the supported CAD system and release, the used storage folder for meta and output data of the application, like captured images or result files, as well as a naming convention for parts and assembly components. Therefore, one KBA can be morphed to various project environments, if the distribution portal supports handling of defined settings and files. Further project-related configurations are CAD systems settings like the overall system accuracy, licenses, a predefined structural layout for part and assembly components and the definition of product data quality guidelines including their verification prescriptions, as illustrated in Figure 3. The presented approach does not focus on the PDM [24] environment but on the illustrated interface between CAD systems and their environment.

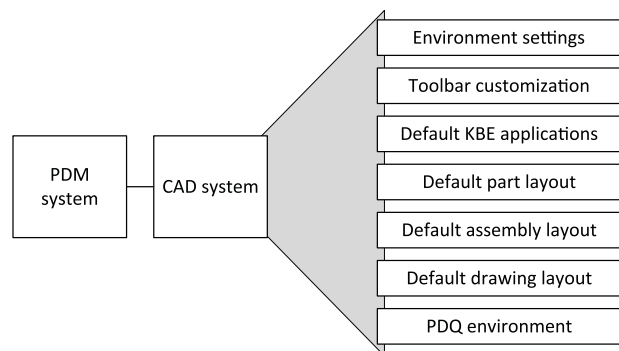


Figure 3. Project-related CAD environment.

2. Global KBE infrastructure – rollout

The common KBE lifecycle in literature consists of the five main parts creation, capturing, formalizing, implementing and activating of knowledge [25], whereby this paper strongly relates to the activation section.

The proposed KBE lifecycle, depicted in Figure 4, covers all mentioned five main parts but it is organized in a different way (DAR):

- Global Directives
- KBAs
- Global Rollout



Figure 4. DAR process.

The global directives and the development environment for the efficient development of KBAs have been proposed in previous research work [22]. In order to provide an overall picture of the DAR process chain, this introduction summarizes the main aspects of these sections.

2.1. Global directives

Starting with the management decision to set up global directives for the development process, a comprehensive CAD design and CAD automation strategy has to be defined. The KBE automation strategy includes the definition of accepted automation methods, programming language(s), programming environment and supported CAD systems. Certainly, these decisions strongly influence upcoming development and operating costs and at the same time the realizable functionality of the environment. The proposed approach can only be implemented successfully with the highest level of KBE automation, as introduced in [22]. Therefore, a development environment using .NET programming language and Visual Studio [18] as integrated developing environment (IDE) is recommended.

2.2. Development of KBAs

Efficient development of KBAs requires a software development kit (SDK) as basic layer. This environment simplifies the KBE development process (especially for not experienced API programmers), reduces repetitive tasks and provides a consistent KBE structure. The SDK consists of core functionalities and abstract class definitions. These base classes are used by surrounding satellites, which implement the functionalities themselves. The mentioned SDK has been introduced in detail in a conference paper [22] in 2015 and is still in the beta phase of development. In industrial applications, more and more KBAs are developed on top of the SDK currently and therefore, the development platform is enhanced step by step. The SDK environment concurrently supports multiple CAD systems and releases within on single KBE application. This simplifies the cooperation of different companies, which often use dissimilar CAD software with different versions or even from different vendors. Especially in the automotive industry, suppliers have to consider varying CAD systems and predefined requirements of their customers. Figure 5 illustrates the effort of maintenance and change management for common development compared with the implemented Multi-CAD approach, which enables the mentioned support of multiple CAD systems [22].

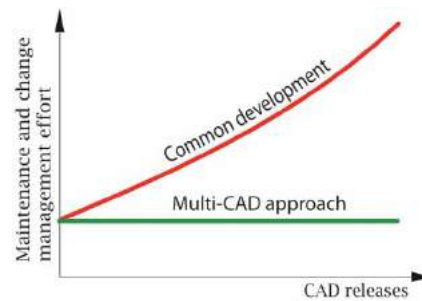


Figure 5. Effort of maintenance and change management over implemented number of CAD releases.

Based on this SDK, various KBAs can be developed in an efficient and consistent way. Each application enhances the SDK and therefore improves the functionalities of the entire KBE environment. The scope of KBAs can reach from relative simple automation routines, e.g. automated product data export or basic geometry features, up to generic sophisticated applications, which are able to strongly improve and characterize development processes. [22] [23]

2.3. Global rollout

The presented approach fulfills four main objectives and closes the proposed DAR loop (Figure 4):

- Management of various engineering projects
- Global deployment, configuration and management of KBAs
- Project related configuration of the CAD system – on demand
- Feedback systems

This global rollout approach requires the development of four interlinked main components:

- Database layout – stores the KBAs and CAD settings, handles access rights . . .
- Management application – allows the adaption and management of the databases
- Information platform – inform all employees, get reasonable feedback . . .
- Client environment – allows the execution of the KBAs on the client machine

Detailed description of the database layout, the management application as well as the information platform can be found in Section 3.

The client application represents the final step of the DAR process and includes the usage of KBAs. Therefore, an adequate client environment is required

(Subsection 3.4). The user field often ranges from project management over product data management disciplines to industrial and engineering designers. This widespread user environment requires a clever and intuitive graphical user interface (GUI) to provide satisfying involvement of all parties. Furthermore, a reasonable feedback and help system is required to support user and to improve the application, the process and therefore the entire DAR lifecycle. In special cases, the client should also provide an administrator or key-user interface for configuration and customization purposes.

3. Realization/validation of the global KBE infrastructure

The cognitions presented in the paper consider that certain preliminary tasks and decisions, related to global directives and development environment, have to be set in advance. Under consideration of this fundamental precondition, challenges can be found in subsequent tasks

of distribution and rollout of the developed KBAs. Not only a secure deployment of the software itself, but also comprehensive and efficient management, maintainability and user feedback handling is required for successful application in daily business. Furthermore, data handling of various project and customer requirements have to be considered in this context. Another challenging task includes the inter-site collaboration considering various development departments all over the world.

The proposed approach covers the mentioned requirements for the deployment of KBAs within a global environment. This includes an efficient management environment, the control of access rights to fulfill security aspects, the handling of project-related necessities and a reasonable feedback method. The approach consists of four interlinked main modules, explained in the next Sections. Figure 6 illustrates the main components and the designed interaction links:

- Database layout - master and site databases

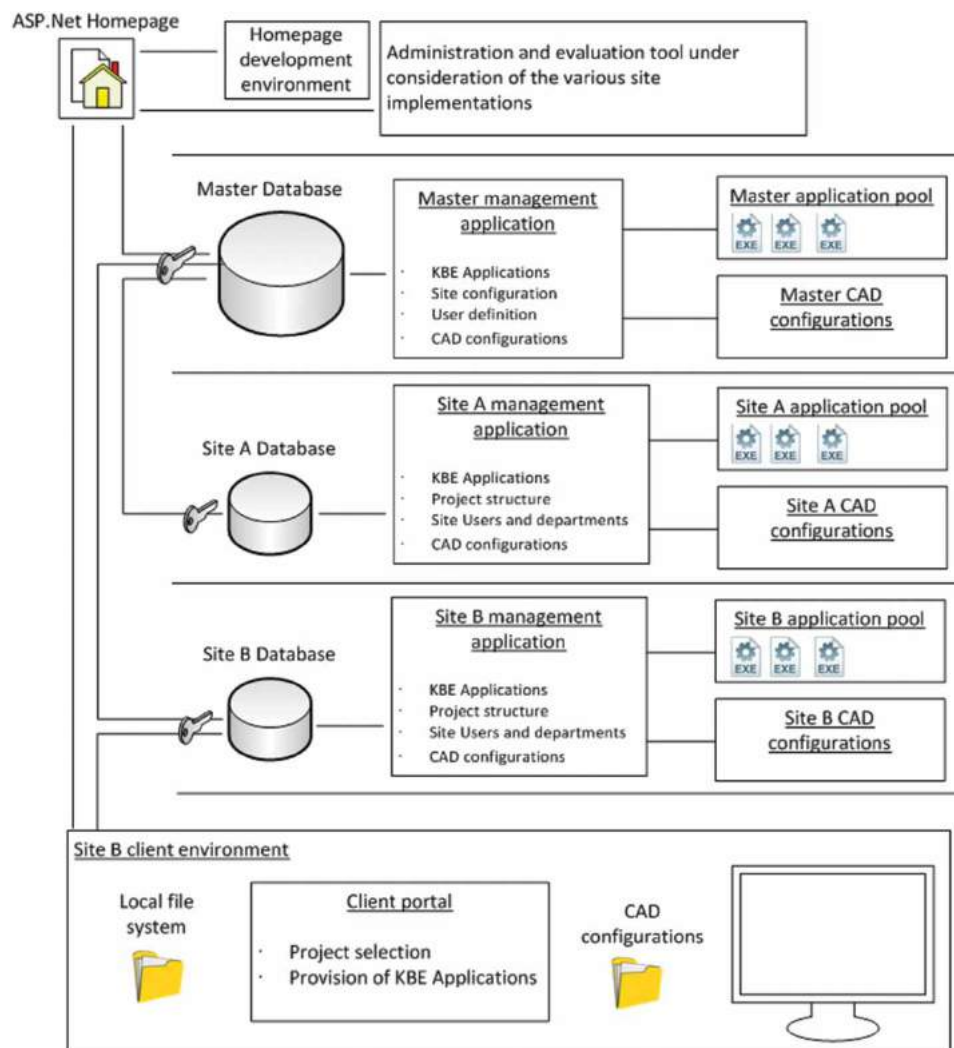


Figure 6. Global distribution concept.

- Management application - master and site implementation
- Information platform
- Client environment

Starting at the bottom of Figure 6, the client environment with local file system and client portal is depicted. The client portal provides the user a list of available projects and KBAs. Based on the selected project, the related CAD system, available KBE applications and further settings are defined for the current session. The client portal receives the required information from the corresponding site database, which is a partial replication of the global master database. The management of database environment is supported by a tailor-made application, which is also shown in Figure 6 – master and site management application. Finally, the approach comprises an information platform, which serves as information, feedback and communication environment. [9] [26]

3.1. Master and site databases

The database layout represents a central part of the approach. The master-site abstraction leads to some advantages regarding security aspects, adaptability and customization options, as well as to performance improvements. Therefore, the master database is partially replicated to the different site servers. This task requires a commercial, open-source or self-programmed Windows-compatible service that depends on the used database.

A benchmark of different relational database systems for the present deployment approach led the authors to the FirebirdSQL [7] database. The most relevant benchmark attributes considered ability for distribution, maximum database/table and row size, available APIs, server-side scripts and user management. The second choice is the Microsoft SQL Server [18] database. There are four different SQL Server editions, whereby one is free of charge - the Express version. This edition has the drawback that the database size is limited to 10 GB and the maximum memory (RAM) is restricted to 1GB. The other SQL Server editions are expensive and do not fit to the open source idea. The selected FirebirdSQL database is open source and based on the SQL syntax. The server can be installed on various operating systems (OS), the API supports a lot of different programming languages and there is a large and innovative community.

In the present application, the database was implemented on a Windows server machine and the VB.NET programming interface was used. The database replication to the different site databases is currently done

by commercial tool IBReplicator [12], but there are also available open source tools like FIBRE [6] or FBReplicator [5].

This replication approach leads to some advantages:

- Security aspects: Only the data transfer (once a day) due to the database replication is sent over the internet. The data transfer from client to server is carried out by the secure local area network (LAN). Furthermore, the site employees have no access to the master database and the entire know-how.
- Adaptability and customization options: Each site is enabled to configure their own environment and this prevents overcrowding and overloading of the master database.
- Performance improvements: Considering a LAN infrastructure, the data transfer rate is no longer limited by the internet connection (some Mbits/s) compared to 1000 Mbits/s using Gigabit-LAN.

The access rights for the projects and available KBAs are related to the unique user name. Considering a mid-size site with about 1000 employees, the manual administration of the user names requires a lot of effort. Therefore, the user names and location information is gathered by the active directory (AD) service of Microsoft. The AD contains a lot of domain-based information, but basically user name, location, email and telephone property is used. The administrator can activate the different KBAs based on the location, the specific project or computer name within the management application (Subsection 3.2). Thus, the database tables “UserID – KBEAppID; ProjectID – KBEAppID; ComputerID - KBEAppID” are adapted and the various site user receive their unlocked KBE application via the client portal. Additionally, the database can be administered manually by using FlameRobin [8], IBAdmin [11] or for example DBManager. [2] [14]

3.2. Master and site management application

A management application enables the configuration and setup of the relational databases. The functionality of the management application depends on registered user and on the specific user roles. If the user is a member of the role “Master”, the entire functionalities of the application are provided. This includes the setup of default CAD configurations, site management and the control of access client rights to the database. Certainly, the master database is responsible for provision of different KBAs within the database [7].

The site database receives a partial copy of the master database accordingly to the defined access rights.

The management suite enables responsible employees to configure their environment according to specific needs. This can concern the definition of a particular project / department structure or site-specific KBE-applications. A mandatory task of the site administrator is the definition of process or user-related access rights, Figure 7.

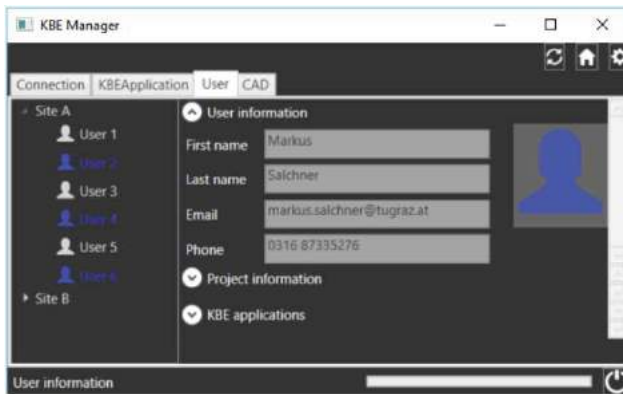


Figure 7. Management application.

Furthermore, the site administrator has to define the project-related CAD environment. To support this task, a default setup for CATIA V5 [3] and Siemens NX [24] was implemented. The first task includes the definition of default installation directory of the CAD system. The client tool – see Subsection 3.4 – uses the command prompt (cmd.exe) or PowerShell to execute the CAD system with respect to the defined functions. For example, a variety of environment variables was set in NX, which point to a directory or file, and thus define e.g. functionality, behavior and available part templates.

In the same way, project- or customer-related product data quality (PDQ) settings can be defined. In case of CATIA, a Q-checker [27] environment is required. In NX, an integrated functionality, called “CheckMate” [24] is used. The tool enables the user to create customized test procedures and to determine, if all sketches are fully constrained, or if there is a specific feature within a geometry group. These relatively simple single tests can be exported to a file (*.dfa), but enhanced tests can require additional script files, like Knowledge Fusion of NX [24] or a VB.NET file.

The consistence of the DAR process (Figure 4) can only be fulfilled by information and data exchange with the master database. Therefore, a copy of each site database on the master base server was created and the corresponding KBAs, CAD settings and PDQ setups were implemented into the global master database.

3.3. Intranet site

An intranet site provides the basis for knowledge distribution and feedback-related functions. The first objective is to inform design engineers about available and planned applications. This includes a short description of each KBE tool, the current development status and a short video clip tutorial. Furthermore, each design engineer can announce personal needs for the listed applications. The development department receives profound feedback regarding the demand of different applications - and so it can adjust the release roadmap. Additionally, the homepage provides an upload site, where engineers can propose new ideas and improvements. This can be done via written text on page and upload of related files, like images, presentations or even existing automation scripts.

In an exemplary project, the homepage was programmed within the Visual Studio environment using the ASP.NET [18] technology. Therefore, the homepage is based on the .NET framework, which allows the implementation of developed methods within the SDK and the use of database functionalities. In this way, the main content of the homepage can be coupled with the mentioned master database. Furthermore, a content management system (CMS) was integrated that allows an easy adaption and integration of the homepage content. The open source CMS tool Umbraco [28] was applied, which is also based on the ASP.NET technology. The site advisor manages the content via a simple online interface, illustrated in Figure 8. The functionality of this interface, the basic layout of homepage and available components are developed in advance. To support the developer, the entire Umbraco environment can be implemented in the Visual Studio environment.

The publication of this homepage requires a Windows server environment including the Microsoft Internet Information Services (IIS) and a Microsoft SQL Server Compact 4.0 installation [18]. The Umbraco kit requires at least this freeware database. Alternatives are Microsoft SQL Server Express or a paid version of Microsoft SQL Server [18].

3.4. Client environment

The final layer (Figure 6, bottom) represents a client environment within a specific company site. Figure 6 illustrates a common client machine for design engineers. The depicted elements include the local file system, where the specific software packages and the CAD programs are installed. The design engineer can work within this native environment. The next block illustrates the client portal. This portal is linked to the site database and allows

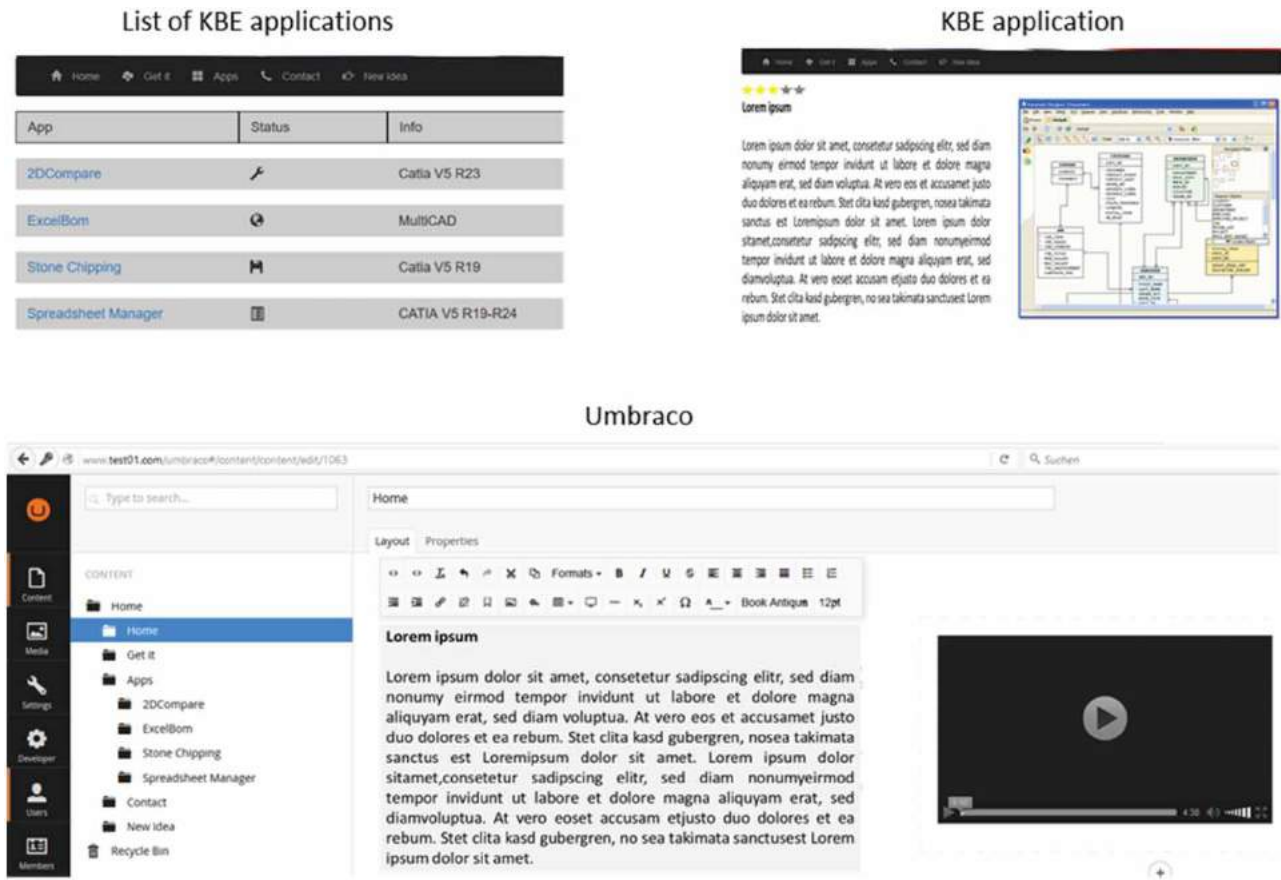


Figure 8. Overview of intranet portal.

design engineers to select the appropriate project. On the one hand, this provides a listing of all available KBAs, which are activated for the selected project or the current user. On the other hand, project-related CAD settings are extracted to the local installation folder, depicted as CAD settings folder. This configuration represents itself as a background task, and so design engineers have not to take care about it. As soon as the configuration is completed, the design engineer can start the CAD application from a button within the client portal, depicted in Figure 9. This click executes the command prompt in background, sets defined CAD project environment variables, as explained in Subsection 3.2, and finally executes the CAD system. In contrast to other approaches, the KBAs are not copied to the local file system. The applications are invoked within the client portal, so there is no local copy of the KBE application on the client machine, and therefore the engineer cannot make a copy of the application or use it in illegal manner.

The client tool stores relevant data of development in the database and supports the feedback idea. This includes a log in case of unexpected failure of a KBE application, but also statistics of used applications including their functions.

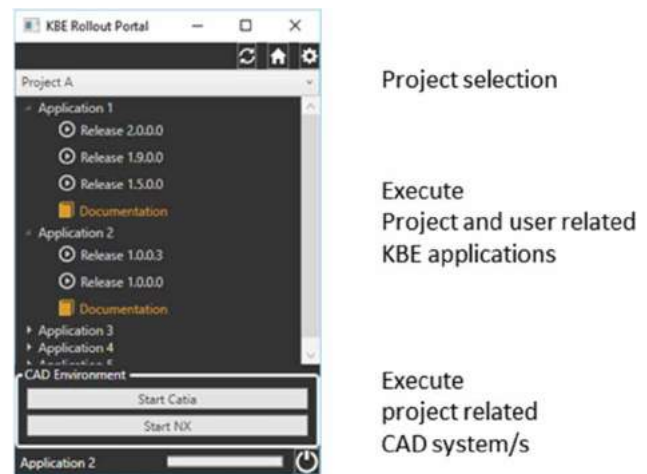


Figure 9. Client environment.

4. Conclusion

The presented approach of a global knowledge-based engineering infrastructure provides a globally available groundwork for the management and distribution of KBAs as well as for the definition of project-related CAD environments including part, assembly and drawing

templates. This comprises an efficient management environment, the control of access rights to fulfill security aspects, handling of project-related necessities and a reasonable feedback management structure. In addition, the development of a database concept, definition of a generic CAD environment and an enhanced server concept is supported and PDQ functionalities are implemented. In global development structures, client computers are beyond the reach of local KBE developer and administrator, which challenges error handling. This requires the possibility to test correct CAD system installations, including available licenses and language settings of CAD and operating systems. The introduced approach has been implemented in an industrial application and is currently in beta testing phase. Intermediate test results show great benefits in terms of increased efficiency, effective development, global provision and management of knowledge-based engineering applications within CAD environment.

One obvious drawback of the present approach includes the initial effort for development and setup. Once these phases are completed, the implementation of the presented method at various companies or departments can be done in an efficient way. A scalable concept including well-structured and documented source code allows further enhancement and customizations.

In the near future, the release phase will be started, supported functionalities will be improved and experience regarding the implementation at various locations will be gained in an industrial application. Finally, the utilization of KBAs in a global enterprise will be supported efficiently.

ORCID

Markus Salchner  <http://orcid.org/0000-0001-6379-0562>

Mario Hirz  <http://orcid.org/0000-0002-4502-4255>

Severin Stadler  <http://orcid.org/0000-0001-9867-0552>

References

- [1] Citrix, <https://www.citrix.de>, access date: 2016-01-23.
- [2] DBManager, www.dbtools.com.br, access date: 2016-02-22.
- [3] Dassault Systemés, <http://www.3ds.com>, access date: 2015-12-14.
- [4] Eclipse, <https://eclipse.org/>, access date: 2015-12-12.
- [5] FBReplicator, <http://www.meta.com>, access date: 2016-02-22.
- [6] FIBRE, <http://fibre.sourceforge.net>, access date: 2016-02-22.
- [7] Firebird, <http://www.firebirdsql.org/>, access date: 2015-04-13.
- [8] FlameRobin, <http://www.flamerobin.org>, access date: 2016-02-22.
- [9] Herrick, D.R.; Tyndall, J.: Sustainable Automated Software Deployment Practices, ACM Annual Conference on Special Interest Group on University and College Computing Services, 2013, <http://dx.doi.org/10.1145/2504776.2504802>.
- [10] Hirz, M.; Dietrich, W.; Gfrerrer, A.; Lang, J.: Integrated computer-aided design in automotive development: development processes, geometric fundamentals, methods of CAD, knowledge-based engineering data management, Springer, 2013, ISBN: 9783642119392, <http://dx.doi.org/10.1007/978-3-642-11940-8>.
- [11] IBAdim, <http://www.sqly.com>, access date: 2016-02-22.
- [12] IBReplicator, <https://www.ibphoenix.com>, access date: 2016-02-22.
- [13] Infosys, <https://www.infosys.com>, access date: 2016-02-22.
- [14] Inmaculada, V. A.; Severin, S.; Hirz M.; Roszbacher P.: The Potential of Relational Database Framework Supporting Knowledge-Based Design, Trends in the Development of Machinery and Associated Technology, 2015.
- [15] La Rocca G.: Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design, Advanced Engineering Informatics, 26(2), 2012, 159–179. <http://dx.doi.org/10.1016/j.aei.2012.02.002>.
- [16] Liao S.: Knowledge management technologies and applications—literature review from 1995 to 2002, Expert Systems with Applications, 25(2), 2003, 155–164. [http://dx.doi.org/10.1016/S0957-4174\(03\)00043-5](http://dx.doi.org/10.1016/S0957-4174(03)00043-5).
- [17] Matrix42, <https://www.matrix42.com>, access date: 2016-02-22.
- [18] Microsoft, <http://microsoft.com>, access date: 2015-12-14.
- [19] Microsoft Developer Network, <http://msdn.microsoft.com/en-us/dn308572.aspx>, access date: 2016-02-22.
- [20] OPSI, <http://www.opsi.org>, access date: 2016-02-22.
- [21] Reddy E.J.; Sridhar C. N. V.; Rangadu V. P.: Knowledge Based Engineering: Notion, Approaches and Future Trends, American Journal of Intelligent Systems, p-ISSN: 2165-897, e-ISSN: 2165-89942015; 5(1), 1–17 <http://dx.doi.org/10.5923/j.ajis.20150501.01>.
- [22] Salchner, M.; Stadler, S.; Hirz, M.; Mayr, J.; Ameye, J.: Multi-CAD Approach for Knowledge-Based, Computer-Aided Design and Applications, 13(4), 2015, 471–483. <http://dx.doi.org/10.1080/16864360.2015.1131540>
- [23] Sanya, I.O.; Shehab, E.M.: An ontology framework for developing platform-independent knowledge-based engineering systems in the aerospace industry, International Journal of Production Research, 52(20), 2014, 6192–6215. <http://dx.doi.org/10.1080/00207543.2014.919422>
- [24] Siemens PLM, <http://www.plm.automation.siemens.com>, access date: 2015-12-12.
- [25] Stokes, M.: Managing Engineering Knowledge – MOKA: Methodology for Knowledge Based Engineering Applications, Professional Engineering Publishing Limited, ASME Press, 2001.
- [26] Tyndall, J.: Building an effective software deployment process. In Proceedings of the 40th annual ACM SIGUCCS conference on User services (SIGUCCS '12). ACM, New York, NY, USA, 109–114. <http://dx.doi.org/10.1145/2382456.2382482>.

- [27] Transcat-PLM, <https://www.transcat-plm.com>,
- [28] Umbraco, <http://umbraco.com>, access date: 2016-02-22.
- [29] Verhagen, W. J. C.; Bermell-Garcia, P.; van Dijk, R. E. C.; Curran, R.: A critical review of Knowledge-Based Engineering: An identification of research challenges, *Advanced Engineering Informatics*, 26(1), 2012, 5–15. <http://dx.doi.org/10.1016/j.aei.2011.06.004>
- [30] Yoo, S. B.; Kim, Y.: Web-based knowledge management for sharing product data in virtual enterprises, *International Journal of Production Economics*, 75(1–2), 2002, 173–183. [http://dx.doi.org/10.1016/S0925-5273\(01\)00190-6](http://dx.doi.org/10.1016/S0925-5273(01)00190-6).