K. Hiekata et al. (Eds.)

doi:10.3233/ATDE190111

Templates for KBE – Their Real-Life Developments and Applications

Jerzy POKOJSKI [a,1], Karol SZUSTAKIEWICZ [b], Łukasz WOŹNICKI [c], Konrad OLEKSIŃSKI [a] and Jarosław PRUSZYŃSKI [a]
[a] *Institute of Machine Design Fundamentals, Warsaw University of Technology, Poland*
[b] *General Electric, Warsaw, Poland*
[c] *Faurecia Grójec R&D Center S.A. Grójec, Poland,*

Abstract. The work presents real, industrial examples of generating, developing and applying UML templates in knowledge modelling for the needs of KBE applications. The applied templates are a synthetic reflection of standard programming constructions used by teams of programmers in specific industrial conditions. Templates are created taking into account both the very individualized specifics of software development, as well as the individualized process of evolution of a specific project area. In this case the construction of the KBE application is implemented as modelling based on carefully elaborated and largely individually tuned templates. The final application of this class is integrated with commercial CAD / CAE systems and other software. The use of the templates allows for a relatively effective, specialized creation of KBE applications. In addition to the skills needed to work with templates, people using this tool must possess clearly articulated, competent and complete substantive knowledge. The article consists of three sections: the first introduces themes to the templates and presents the process of initialization of this tool class in industrial conditions; the second section shows the use of modelling in the process with the gradual evolution of the modelled knowledge; the third section concerns the application of a strategy based on the rational development of a representative set of templates. The examples presented are based on real industrial implementations conducted over a period of about 10 years.

Keywords. Knowledge Based Engineering, Templates

Introduction

The work presents real, industrial examples of generating, developing and applying UML templates in knowledge modelling for Knowledge Based Engineering (KBE) applications [1, 2, 3, 4]. The applied templates are a synthetic reflection of standard programming constructions used by teams of programmers in specific industrial conditions.

The basic problem with the construction of an advanced, comprehensive model of the design process is the selection of knowledge representation and implemented tools. Both: tools available in the KBE modules of CAD / CAE systems, as well as strictly programming tools (both based on algorithmic and declarative approaches) can be used, and can also be integrated with CAD / CAE systems [5, 6, 7, 8].

[1] Corresponding Author, Email: jerzy.pokojski@pw.edu.pl.

In general, the construction of the KBE application is a labor-intensive task. It requires to build properly structured geometric models by adding the relevant elements of knowledge.

In the process of creating KBE applications, both: the product and the design process are usually modelled.

By using Pahl and Beitz theory [9], the following phases may occur in the design process: clarification of design conditions, conceptual design, embodiment and detailed design. Each of these phases has its own specificity, which can directly and indirectly influence the form of the KBE tools used [1, 2, 3, 4]. An equally important factor affecting the applied KBE solutions may be the specificity of the field and the tools used to support engineering works. All in all, this gives quite a broad spectrum in terms of concepts and the processes of KBE application creation. In practice, the most frequently chosen and applied solutions are those that are best mastered and have a relatively low level of implementation complexity [10]. KBE applications developers are very eager to use proven and well known solutions presented formally or informally. The standard form of actual experiments in these cases are templates. The KBE application is created on the basis of using specific, fulfilling requirements, previously created and reusable templates - programming patterns [1, 2, 3, 4, 5, 10, 11].

The proposed patterns rarely relate to complete product models and processes. Usually, they are smaller tasks and design problems, called partial ones. They may include templates for the corrective or generative creation of design variants, templates for studying relations between objects, templates for modelling the structure of designer preferences, etc. If the template resources are appropriately suited to the modelled tasks and are appropriately numerous, the KBE application development process largely boils down to the choice, application and integration of some of them.

The beginning of the computer model of each design process is specific knowledge [7, 8, 10, 12, 13]. This knowledge is in the possession of designers, and is owned and used by them. Often, some fragments of this knowledge are articulated and formally written, close to existing patterns. Having ready-made patterns, it is possible to create the currently needed KBE application.

The following sub-chapters show cases of using a template-based solution:

- the first introduces the subject of templates and presents the process of initializing this class of tool in industrial conditions (example based on modelling a spiral stair design process; the templates generated in this process were used again to model the design process of a car gearbox) [1, 2, 3, 4, 10],

- the second shows the use of template-based modelling in a process with the gradual evolution of modelled knowledge and used templates (example of industrial modelling of knowledge in the field of creating FEM models used for calculating car seats) [14],

- the third concerns the application of a strategy based on a rational, conceptually and formally broad development of a representative set of templates (an example of industrial modelling of knowledge in the field of a conceptual and detailed design process production lines for metal forming) [1, 2].

1. Initialization of the process of creating templates in industrial conditions

The original task was to build industrial software to support the design process of spiral stairs (figure 1) [1, 2, 3, 4, 10]. This software served as a template provider. The obtained set of templates was treated as standard and representative for the considered software class.

Figure 1 also presents a conceptual, fragmentary description of the solution templates used in the construction of a system supporting the design process of spiral stairs. The set of templates presented was the basis for creating other, subsequent applications. The solutions included in the proposed templates were to a large extent universal. The proposal included various implementation concepts depending on the phase of the design process, visualization standards, the scope of interaction with the user, integration with other software, etc.

Templates included in the table in figure 1 were used to very efficiently create KBE applications for the design process of a car gearbox (right section figure 1). Practically, it was a replica of work created earlier on the basis of classic programming solutions. Creating the replica using templates was done very quickly (about 20 times faster than the original).

The approach presented above does not allow to directly capture the dynamics of the development process of a specific template. This is practically done by creating, modifying and multiplying consecutive versions of templates.

Other solutions to the problem indicated above are presented in the following chapters of the paper.

2. Gradual evolution of modelled knowledge and applied templates

The chapter presents an example where the modelled templates undergo a gradual metamorphosis by means of strictly defined, previously planned modifications. In certain application classes, this type of approach proved to be sufficient. This introduces a predetermined, permissible multi-state.

The use of templates in the construction of the characterized KBE tool class significantly improves their development. Implementing them does not only accelerate the creation process, but also greatly facilitates the introduction of changes and improvements in existing tools. The use of templates, however, also requires appropriate experience from the person building the new tool. It is necessary to see the places where the templates can be used, and to decide which template will be the right solution in a given situation. The experience and ease of using templates usually come with increasing practice.

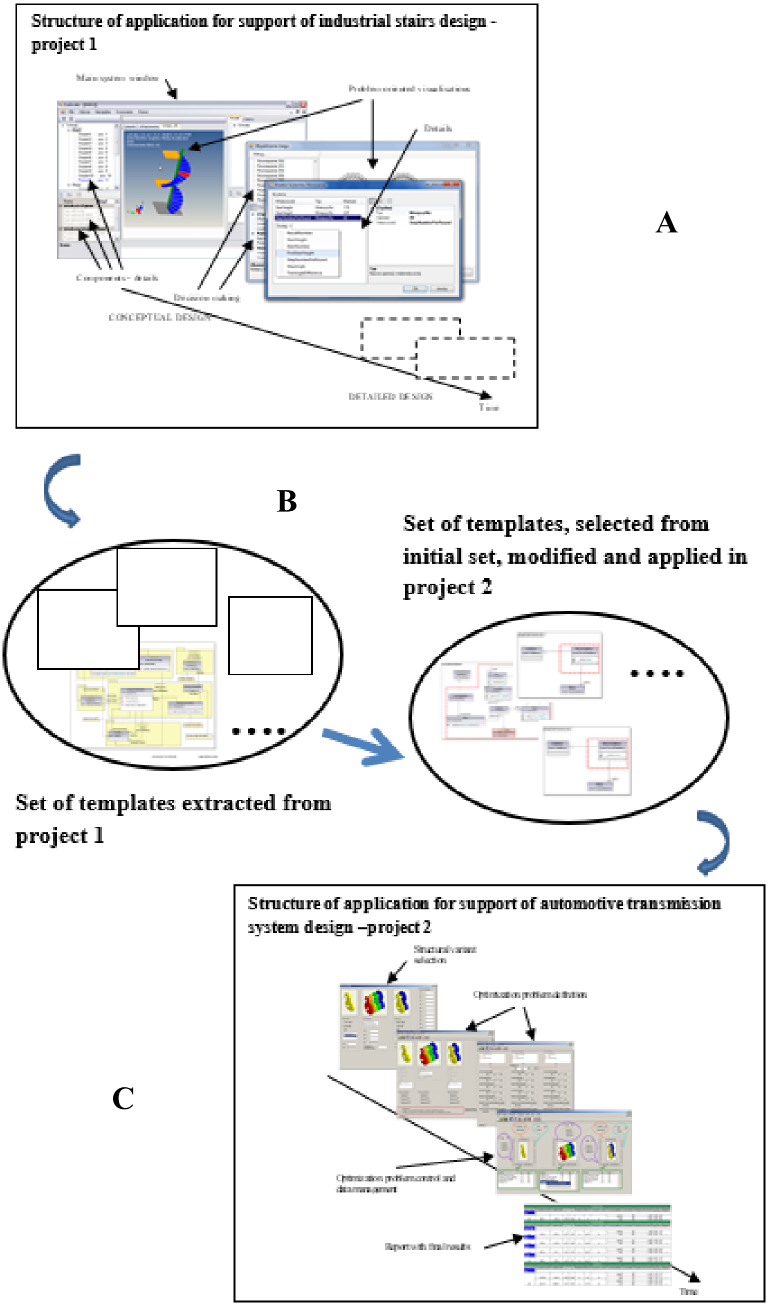
Structure of application for support of industrial stairs design - project 1
Main system window
Problem oriented visualisations
Details
Components - details
Decision making
CONCEPTUAL DESIGN
DETAILED DESIGN
Time
A
B
Set of templates, selected from initial set, modified and applied in project 2
Set of templates extracted from project 1
Structure of application for support of automotive transmission system design –project 2
Structural variant selection
Optimization problems definition
Optimization problems control and data management
Reports with final results
Time
C

Figure 1. A. The process of creating a KBE application supporting the process of designing spiral stairs and separating a set of templates. B. Typology of obtained templates C. The process of using the obtained set of templates in a new KBE project.

As part of the work, a concept of a system supporting the building of KBE tools with the use of templates was created. In addition, a system was designed to support the acquisition, storage and management of engineering knowledge necessary to build KBE tools. The concept assumes that the system bases on available knowledge and suggests the best substantive solutions. Access to system resources varies, depending on the role of the user. The following groups of users are taken into account: 1) knowledge providers - experts, specialists in a given field and engineers working on a given issue every day, 2) knowledge engineers - experts in the field of modelling engineering knowledge and creating KBE systems. Depending on the role played, users have access to various data and functionalities.

It was assumed that the knowledge is stored in textual and multimedia forms with the possibility of using and creating more complex structures and forms of representation.

The system was divided into 3 layers: the knowledge acquisition layer, the knowledge management layer, and the layer generating the formal model of the product / process, modelled using the knowledge defined in the two previous layers.

The knowledge acquisition layer (informal model) is used to collect and modify engineering knowledge on a given topic. This knowledge can be written in a simple, unstructured form, and can also be combined with other elements. Separating this layer enables the user to create a repository of knowledge that contains all the elements of knowledge describing a given product or process and, if necessary, allowing for the modification and extension of stored knowledge with new elements. Each knowledge element has its own identifier with which its evolution process can be followed, and in the long term the knowledge development of a given product / process can be observed. Knowledge providers are experts and specialists in a given field, as well as engineers who have constant contact with real problems in their everyday work.

The knowledge management layer (formal model) is used to organize the knowledge gathered in the previous layer. The task of the user is to describe all elements with elementary primitives that contain as little information / knowledge as possible, and then connect them by means of a relationship. The result of such an operation is a structured representation of knowledge, with defined and appropriately named connections between particular knowledge elements (figure 2).

In the third layer, the product / process model is generated on the basis of the previously defined knowledge model and uses the attached template database (figure 3). Pre-defined relationships between knowledge elements are applied to match the most appropriate template. (The authors are working on the concept and implementation of an editor designed for such functionalities.)

The generated model fully corresponds to the previously defined knowledge model. There is also the possibility of making changes to the generated model in a manual manner by the person responsible for the development of the tool. Additionally, classes that build this model have their equivalents in the knowledge model, thanks to which it is possible to quickly find the place where the application implements a selected fragment of knowledge. The system controls both models continuously, and any change to one of these models is automatically noticed and asks for appropriate changes.

The application of the system provides an easier control of the knowledge management process of a given product and, consequently, the management of the application that implements this knowledge. The system also allows to trace changes that have occurred, both in the application and in the modelled and stored knowledge.

The proposed approach has its source in a work [14, 15] concerning the process of knowledge modelling in KBE applications to support the calculations of car seats using the FEM method. Generally, it refers to the concept of MOKA [5].

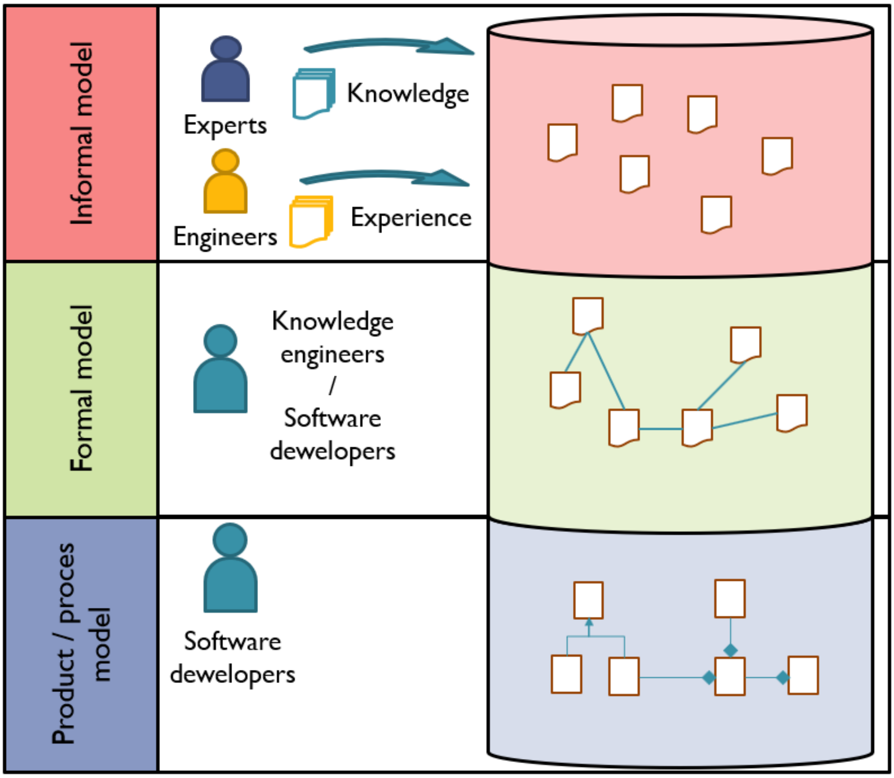

Figure 2. Functioning of individual layers of knowledge representation.

3. Rational development of the set of templates

The chapter presents one more way of showing the development dynamics of a specific template. The proposed approach uses the offer made in the MOKA project [5]. The MOKA project assumes two stages of the model's operation: informal and formal. Both models refer to the same activity (or set of project activities) and they are like two stages of development of the same issues. In the first case there the "informal" intake - direct cooperation with a human / designing person dominates; in the second case a "formal" version allows for easy automation of the process. Additionally it is possible to build a template that is at the same time a record of both these elements, with the template the different states of the “informal” class and also the various states of the “formal” class can be captured. It was assumed that each time the template has the same structure - the development of the resources which are stored in it are consolidated.

The classic UML templates were created as a direct development of the programming tools used earlier. With their help ready-made complexes of knowledge or information can be operated providing that the reality and tools are mutually aligned [5, 11]. Meanwhile, many elements that constitute important project knowledge go beyond this scheme, develop in a different way to the earlier plan and prediction [12, 13]. The resources of the used knowledge may have new, different representations. They may cooperate with the external environment through scenarios difficult to predict - we are already observing such processes. Therefore, the concept of a template based on the entire MOKA project approach (scenario of transition from informal to formal) should be provided with information and tools allowing for quick cooperation with these external changes. The issue is presented in the next sub-chapters.

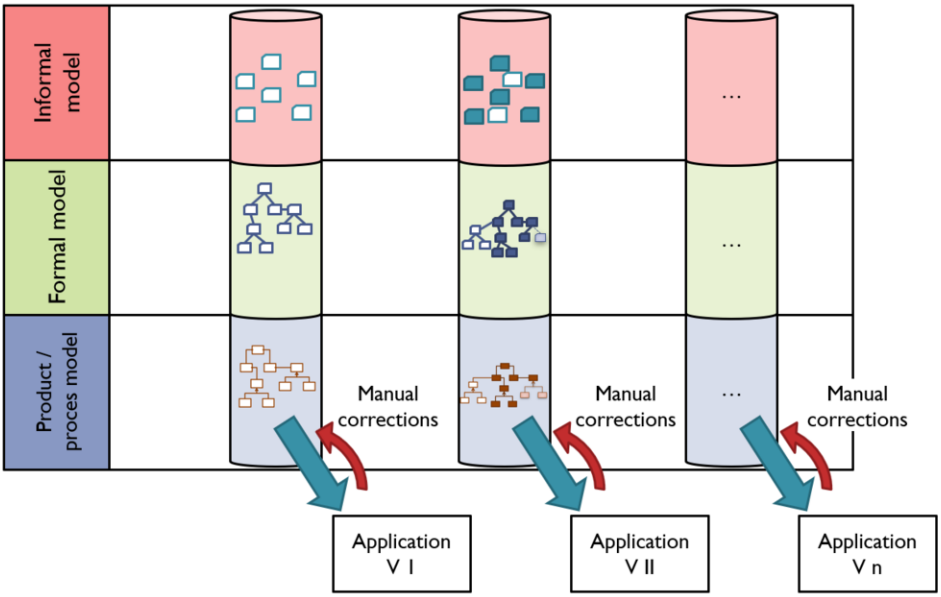

Figure 3. Diagram of the processing process implemented based on modelled knowledge.

3.1. Example. Introduction

This chapter presents templates as well as the process of their development which was observed in cooperation with a company involved in the design and manufacturing of dedicated production lines in the area of steel sheet processing [12, 13].

In the initial period of the development of this approach, dedicated tools were developed to collect and use design knowledge. Some design templates were identified (mainly for relatively uncomplicated issues) and then they were formally described and developed. Next, actions were taken to build highly formalized templates covering all the project tasks that are carried out in the company.

With time, there was an increase in the formalization and the degree of advancement of computer techniques and tools being used. The proven and used methods and solutions from previous periods were transferred, developed, formalized and used in current projects. Due to the small size of the project team and the lack of external support, only those resources were developed that were needed at a given time, or their use was planned in the horizon of 1-2 years. The variable demand (depending on the tasks currently being performed), as well as the class of issue (determined in the internal classification system), resulted in a significant differentiation from the development of tools for individual application areas. The achieved status and the desired state of the whole system are presented in the next subsections.

3.2. Example, Actual state

Currently, there are many templates in the described company [12, 13]. Their advancement varies – both as a substantive and as a tool. Many of them are already full KBE systems.

Template classes and their complexity are basically correlated with the product hierarchy (they are: Knowledge Element, Part, Sub-System, System, Production Line).

The construction of the KBE application is characterized by considerable labor intensity. That is why the templates in the group Knowledge Element are relatively the most numerous. Due to the fact that the knowledge that is used there is not too extensive, they are also the most complete.

Group templates that are higher in the hierarchy very often contain templates from lower groups. As a consequence they must be managed with suitable mechanisms which contain control instructions for reference templates from other classes. Often this is related to the handling of exceptions. Unforeseen circumstances arise, and they have to be eliminated by abandoning them or by editing specific parameters of specific features.

3.3. Example, Goal of development

Currently, using the resources in the company, it is possible to implement various design processes. However, the transition between the individual stages does not always run smoothly. The templates are often not integrated with each other, some parts of the processes are not formally modelled, while others (functioning in the form of descriptive information) are the object of mental processing of project team members.

The company aims at an evolutionary investigation into generally more advanced solutions. The final goal is to build generic and complete templates consisting of lower-order templates for manufactured production lines. In addition, templates from a given row / layer and from different layers must be integrated with each other. They must be properly managed so that they can be suitably manipulated, modified, combined and integrated while maintaining the development history with the background of the decisions behind it. Appropriate mechanisms must manage the elements / templates contained in individual layers. They, in turn, are managed by Cross-Layer Management Mechanisms (Templates), which are also likely to take the form of templates. Their additional task is also responsibility for communication with the user. Figure 4 presents the concept of construction of the described approach.

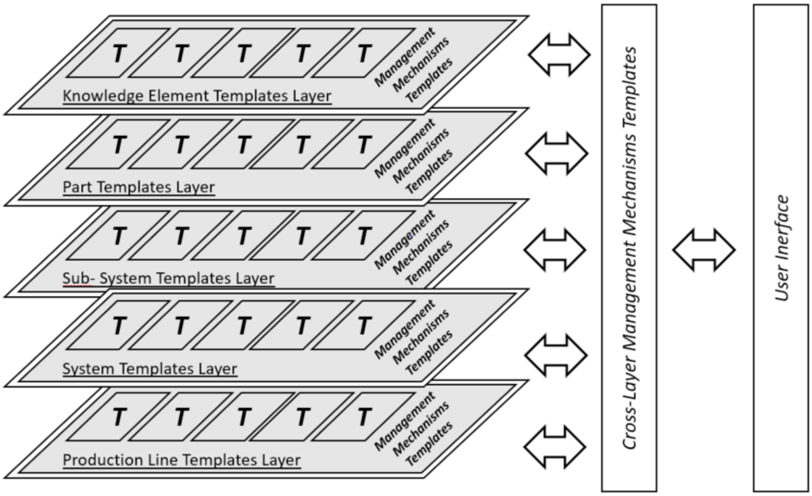
T
Management Mechanisms Templates
Knowledge Element Templates Layer
Part Templates Layer
Sub- System Templates Layer
System Templates Layer
Production Line Templates Layer
Cross-Layer Management Mechanisms Templates
User Inerface

Figure 4. The concept of templates management mechanisms.

4. Conclusions

This work presents several solutions related to attempts to broaden the application of template ideas while using and creating KBE tools in industry The work was based on classical solutions, under the influence of industrial realities. It lead to attempts to introduce new concrete problems both knowledge development and the automation tools that reflect it, and at the same time take into account their real industrial dynamics.

Discussed issues have also their statics and implementation side. There are various basic achievements in the so-called pure form. There are more partial achievements in development, mainly in the area of examples 2 and 3.

5. References

[1] J. Pokojski, M. Gil, M., K. Szustakiewicz, Engineering Knowledge Modelling in Design. In Jerzy Pokojski, Shuichi Fukuda, Józef Salwiński (eds.): *New World Situation: New Directions in Concurrent Engineering, Proceedings of the 17th ISPE International Conference on Concurrent Engineering (Advanced Concurrent Engineering)* , Springer-Verlag, London, 2010, pp. 257-266.

[2] J. Pokojski, M. Gil, K. Szustakiewicz, Extended KBE in Mechanical Engineering- discussion of concepts, In Daniel D. Frey, Shuichi Fukuda and Georg Rock (eds.): *Improving Complex Systems Today: Proceedings of the 18th ISPE International Conference on Concurrent Engineering (Advanced Concurrent Engineering)*, Springer-Verlag, London, 2011, pp, 267-274.

[3] J. Pokojski, J., M. Gil, K. Szustakiewicz, Extended KBE in Mechanical Engineering- discussion of solutions, In Daniel D. Frey, Shuichi Fukuda and Georg Rock (eds.): *Improving Complex Systems Today: Proceedings of the 18th ISPE International Conference on Concurrent Engineering (Advanced Concurrent Engineering),* Springer-Verlag, London, 2011, pp. 275-284.

[4] J. Pokojski, K. Szustakiewicz, Extended KBE – Scenario of an Application Development. In J. Stjepandic, G. Rock, C. Bil (eds.): *Concurrent Engineering Approaches for Sustainable Product Development in a Multi_Disciplinary Environment. Proceedings of the 19th International Conference on Concurrent Engineering,* Springer-Verlag, London, 2012, pp. 291-302.

[5] M. Stokes, *Managing Engineering Knowledge, MOKA – project*, Professional Engineering Publishing Limited, London, 2001.

[6] J. Sobieszczanski-Sobieski, A. Morris and M. J.L. van Tooren, *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*, First Edition. John Wiley & Sons, Hoboken, 2015.

[7] W.J.C. Verhagen, P. Bermell-Garcia, R. E.C. van Dijk, R. Curran, A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics*, Volume 26, Issue 1, January 2012, pp. 5-15.

[8] J. Stjepandic, W.J.C. Verhagen, H. Liese and P. Bermell-Garcia, Knowledge-Based Engineering, In: J. Stjepandic, N. Wognum, W.J.C. Verhagen, *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, Springer International Publishing, 2015, pp. 255-286.

[9] G. Pahl, W. Beitz, and al., *Engineering Design: A Systematic Approach*, Springer-Verlag, Berlin, 2007.

[10] M. Gil, J. Pokojski, S. Skotnicki, K. Szustakiewicz, *Computer Support of KBE Applications Development in Mechanical Engineering*, Institute of Machine Design Fundamentals, Warsaw University of Technology, 2011, (In Polish)

[11] S. Fenves, et al.: CPM: A Core Model for Product Data, *Manufacturing Systems, Integration, Division, National Institute of Standards and Technology, Gaithersburg, USA Available at: http://www.mel.nist.gov/msidstaff/sriram*. Accessed on: March 10th, 2007.

[12] J. Pokojski, K. Oleksiński, J. Pruszyński, Knowledge based processes in the context of conceptual design, *Journal of Industrial Information Integration, In Press,* https://doi.org/10.1016/j.jii.2018.07.002 (it will appear)

[13] J. Pokojski, K. Oleksiński, J. Pruszyński, Conceptual and Detailed Design Knowledge Management in Customized Production - Industrial Perspective, *Journal of Computational Design and Engineering,, In Press,* https://doi.org/10.1016/j.jcde.2019.02.004 (it will appear)

[14] J. Pokojski, Ł. Woźnicki and M. Dębniak, Design Rationale, Knowledge Based Engineering and Knowledge Management in Large Supplier Company, *Advances in Transdisciplinary Engineering, Vol. 7: Methods for Social Innovation of Industry 4.0,* IOS Press, Amsterdam, 2018, pp. 1054-1063.

[15] S. Szykman, R.D. Sriram, Ch. Bochenek, J.W. Racz, J. Senfaute J**.,** Design Repositories: Engineering Design's New Knowledge Base. *IEEE Intelligent Systems,* May/June, 2000, pp. 48-54.