General KBE model with inheritance and multi CAD support

T L Tiuca\textsuperscript{1}, C Rusu\textsuperscript{1}, S Noveanu \textsuperscript{1} and D Mandru\textsuperscript{1}

\textsuperscript{1}Mechatronics and Machine Dynamics Department, Technical University of Cluj Napoca, Romania

E-mail: Teodor.Tiuca@mdm.utcluj.ro

Abstract. Knowledge-Based Engineering (KBE) is a research field that studies methodologies and technologies for capture and re-use engineering knowledge. The primary objective of KBE is to reduce time and cost of product research processes and/or product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and re-using engineering knowledge. Every CAD System includes KBE Tools. The power of these tools is incremented by the use of external high level programming language. The model presented in this paper has the aim to reduce times and costs of particular KBE Models development, by programming inheritance concepts and also the multi CAD Support. The model is implemented through a C# application that is also presented.

1. Introduction

Knowledge-Based Engineering (KBE) is a research field that studies methodologies and technologies for capture and re-use engineering knowledge. The primary objective of KBE is to reduce time and cost of product research processes and/or product development, which is primarily achieved through automation of repetitive design tasks while capturing, retaining and re-using engineering knowledge [1], [2], [3], [4].

Every CAD System includes KBE Tools. The power of these tools is incremented by the use of external high level programming language.

The aim of this paper is to introduce the concept of KBE Management. Why KBE management? Because the development of KBE tools has also its time and costs for developing and a lot of tasks are very repetitive.

We are focusing on KBE tools that execute loops. That is, repeat the same task for a given number of times over the CAD System. Basically any simulation process is a Loop Process, but there are also a lot of other processes which can be modeled as Loop Processes.

2. KBE loop processes dedicated to a unique CAD program

The “classical” Loop Process dedicated to a unique CAD program has the structure presented in figure1.

After the initialization and the data input, the loop will be executed, finally the results are presented. Programming a KBE tool for executing a Loop Process means programming the data input, the loop, and the data output, define the step body and interact with the CAD System [6].

Each time we need to define a new tool is necessary to define almost all, and to change the target CAD System means to build other application [7].
3. The model structure for managing KBE processes
We propose a hierarchical model for loop KBE processes consisting of three levels:

![Diagram of the model structure for managing KBE processes](image)

**Figure 1.** “Classical” KBE Loop.

**Figure 2.** Inheritance model.
The structure is based on the inheritance indicated by the blue arrows. Inheritance is one of the main characteristics of object-oriented programming. Inheritance allows creation of classes based on the behavior of other classes and extend it with new features or specialize it for specific task.

- The first level is the Programming Model, which is responsible to execute the loop. The virtual methods of this model will be overridden by the inherited models.
- The second level is the KBE Model, which deals with the engineering tasks. It extends the Programming Model and it defines the initialization, the loop body and the final tasks.
- The third level is the CAD Model. It extends the KBE Model and it overrides the virtual methods to interact with the CAD Application.

4. The class inheritance
Using this structure it is possible to use the same application for interaction with various CAD systems.

![Inherited Classes Diagram]

Figure 3. Inherited classes.

The arrows are indicating from the derived class to the base class. The Programming Process is executing the loop and calling the virtual methods which are redefined at next level. The KBE Process is defining what is to do in each phase of the process and is calling virtual methods for interacting with the CAD Systems. The CAD Processes are implementing the final methods.

5. Application implementation
For testing this ideas we focused on two CAD systems:
- Dassault Systemes CATIA
- Autodesk Inventor

The application is written in C#, with the Microsoft Visual Studio 2015 Environment. Using the Code Map facilities of VS, we can see the architecture of the classes. The architecture is similar in both cases, the base classes are the same. At this moment chose the CAD System is a compile-time decision. Our further intentions are to realize an application which can allow the user to choose the CAD system at execution time.

Tools.dll is a general utilities library. The base Programming Model is inherited by the KBE Model, and this is inherited by the two dedicated CAD Models. The application is the exe file. The inheritance is marked by the gray arrows with dotted line.
6. Implementation example
Established the bases, we implemented an application for robot inverse kinematics based on assembly restrictions used by any CAD Program. Having a 6 DOF Robot and imposing the position of the end element, one can read the values of the angles in all the 6 joints. By geometrical constructions the position of the end element can be established as a mono-parametric one (displacement over 3D curve and normal imposed, e.g.)

The really interaction with the CAD System is setting a parameter for the position of the end element of the robot and reading the values of the joint parameters. To do this, we need a unified method for accessing parameters. The component-name strategy is good for the two CAD systems in study. That means, identify the component with given name and find the parameter inside. The set and get methods for the parameters are similar in both systems.

To do the job the Update method is to be called after the change of the parameters value. With the same application and two defined classes, one for each CAD System, that are implementing the functions SetParameter, GetParameter and Update, we can simulate the process in both CAD Systems.

Figure 4. Code Map for CATIA.  
Figure 5. The Code Map for Inventor.  
Figure 6. The Catia Loop Process.
7. The User Interface
The actual state of the user interface is designed to show the functional differences between the three models, the Programming model, the KBE model and the CAD Model.

For the inverse kinematics example the interface presents as output results the graphical variation of joint angles. Of course, the graphic results are the same in both cases.

Figure 7. Application Interface.

The application and the interface is prepared for future works of our workgroup regarding KBE management. It is open to any other engineering applications and/or CAD System.

8. Conclusions
KBE management should be for KBE what KBE is for CAD. That means a methodology to reduce cost and times. Using the inheritance process one application can reuse programming logic and engineering knowledge to interact with different CAD systems. By using efficient programming and inheritance, switch to another CAD system consists only in redefining the basic functions.

9. References
[1] Calkins D E, Egging N and Scholz C 2000 Knowledge-Based Engineering (KBE), Design Methodology at the Undergraduate and Graduate Levels International Journal Engineering 16 pp 21-38
[2] Dingyong T, Yizhen T, Long J and Zheng C 2009 Application Research of Knowledge Management in R&D Enterprise Project Management Information Management, Innovation Management and Industrial Engineering International Conference 4 pp 447
[3] La Rocca G 2012 Knowledge based engineering: between AI and CAD. Review of a language based technology to support engineering design Advanced Engineering Informatics pp 159-179
[4] Penoyer J A, Burnett G, Fawcett D J and Liou S-Y Knowledge based product life cycle systems: principles of integration of KBE and C3P Computer-Aided Design 32(5-6) pp 311-320
[5] Stjepandić J, Wognum N and Verhagen W JC 2015 Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges (Springer, Technology & Engineering)

[6] Tiuca T 2015 Proiectare bazata pe cunoastere- Referat II Doctorat (Cluj- Napoca: Universitatea Tehnica din Cluj-Napoca)

[7] Tiuca T, Rusu C, Noveanu S and Mândru D 2015 Study Of A General KBE Model For Mechanical Systems Analysis Scientific Bulletin Series C: Fascicle Mechanics, Tribology, Machine Manufacturing Technology 29 pp 86-89