INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 – 18, 2005

EXTENDED CAD MODEL

Nenad Bojcetic, Neven Pavkovic, Davor Pavlic

Abstract

The main objective of this work is to propose an information model for the design knowledge representation. The proposed information model will be used as a basis for the development of a program system to support the process of creation of the computer model of a product. The developed program system will be able to relate the Feature Based Design (FBD) CAD model of the product with the knowledge used in its creation. The FBD model of the product and the captured design knowledge make an Extended CAD model (ECAD).

Keywords: Feature-based modeling, design knowledge, engineering design, STEP

1. Introduction

Most CAD systems are primarily used for the creation of the computer representation of a real model of the product as well as for the creation of technical documentation. Unfortunately, CAD systems are focused especially on the problems of handling the graphic representation of the product model with limited support for the interaction with the knowledge databases. During the process of development and improvement of CAD systems, the emphasis is mainly put on techniques and technologies for the improvement of the representation and manipulation of graphic models. In most engineering domains, especially in the field of mechanical design, geometric representation is not the only form of information about a product that concerns a designer.

The designer has to consider the information about used materials, constraints of the available technologies, transport limitations, exploitation of the product, etc. In addition, we are witnesses of an increased complexity of products and of a more rapid product development resulting in enormous increase in the amount of knowledge one must deal with. In order to overcome these difficulties some companies are focused on the more efficient usage of internal and external knowledge, reusability of existing components and design solutions, collaborative design and usage of artificial intelligence methods [1]. The main objective of this work is to develop a program system to support the process of creation of the product computer model that will be able to relate the Feature Based Design (FBD) CAD model of the product with the knowledge used in its creation. The proposed solution is an Extended CAD model of the product containing the geometric representation of the model of the product and the knowledge required for its creation.

2. Design knowledge

The nature of the design process and the complexity and variety of the knowledge used during the process of product creation require a flexible and robust model for the representation and handling of the design knowledge [2]. One of the important issues in the design process is the process of knowledge gathering and processing. The decisions that the designer makes during the design process and the used design solutions depend on the knowledge available to the designer at the time.

The study of design knowledge was conducted in the Koncar Power Transformers company that produces electrical power transformers. During the six months period, the information flow and the nature and sequence of decisions that the designer derives on the available information. The needs and specifics of the company's traditional design process, where established by a series of exhaustive unstructured interviews performed with the designers.

Production of complex investment products such as power transformer is time-consuming process, mostly consisting of several phases. The contracting and functional design phase, took most of the time in order to capture client's needs and/or specific regional standards. In traditional transformer development concept the embodiment design, and manufacturing phases follow after in the sequential sequence. In order to shorten the total product development time, most of the phases are now running concurrently, with significant overlaps. The most critical situation arises from the fact that embodiment design phase is launched before the project details are fully elaborated with the final (from contract derived) specification. The design process usually starts with minimum data needed. Occasionally, there could be a rise of requests for minor or major changes to the basic concept after design is more or less done. Those changes, are often unpredictable, initiating resets in product development that may cause partial or complete redesign with time consuming revisions in the corresponding documentation.

Usage of advanced 3D CAD-Systems, which are necessary tools that support the modern product development, has shorten the design process time and improved handling of even major changes. However, resolving those changes is often to complicate for novice designers and senior designer involvement is required since the design rationale is not obvious from the CAD model. Further more, this kind of interventions could be time-consuming and not always totally successful resulting with rebuilding or restructuring the product's CAD model. Based on this approach, there is an argument to build application that will help the designer to handle not only simple but even rapid changes in product design. This is particularly interesting for the products that may be classified as variant designs, i.e. the designs where the product functionality, acting principles and structure are within well known boundaries.

The FBD CAD model of power transformer was built using the ProEngineer system [3].. The FBD model was created in such a way that it will be possible to alter the geometry (create variants) without compromising the integrity of the model. This was achieved by the implementation of design logic and knowledge, outside of the geometry model, itself in the external program system. That knowledge was later reintegrated into the FBD model through the usage of the Extended CAD system.

The focus of our study was primarily on the process of the reuse of existing design solutions and the creation of variant design models. Based on the analysis of the information generated during the design process [3], the design knowledge is grouped as follows (Figure 1):

- Design knowledge;
- Operational knowledge;
- Procedural knowledge.

Design knowledge represents a knowledge generated or used during the design of the product. Virtually every aspect of the FBD model can be represented using elements of design knowledge. In a pilot project, we were focused on the following essential elements of the FBD model:

- Dimensions – define nominal geometry;
- Parameters – additional attributes used in model definition;
- Components – elements that constitute assemblies.

Operational knowledge is an extension of the design knowledge and it contains the information about users: name, organization, role, levels of security for each user and for a defined model (viewing and modifying), time stamps (when the model was created or modified, when particular design knowledge element was created or modified, by whom, whether the modification or creation is approved, and if so, who approved it).

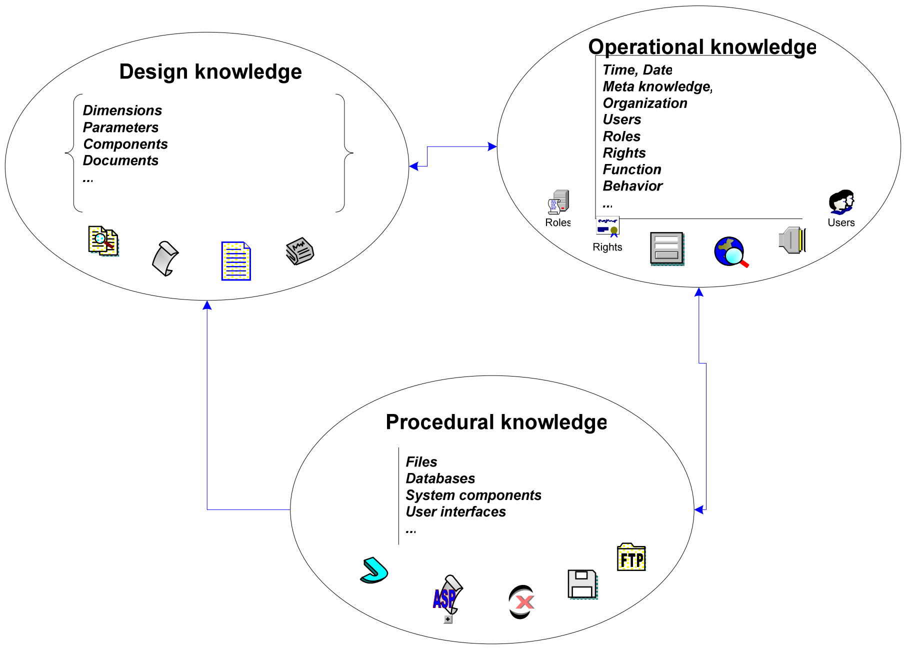

Figure 1. The design knowledge structure

Procedural knowledge relates to the process of management and manipulation of the system data and also to the information about external data sources (locations of databases, connection strategies, locations of external libraries, and initialization files). Procedural knowledge also defines how to interpret design knowledge item data and ensures syntactically correct data input.

Design knowledge is further divided into knowledge items (Figure 2) that are considered to be the carriers of certain types of knowledge. The knowledge items are as follows:

- Elements (item defining the version of the product or product part);
- Rules (item defining the sequence of the applied constraints in a particular version);
- Documents (item defining the documents relevant for the product);
- Applications (item containing the information about required external applications);
- Preferences (item defining material properties, units, and parameter boundaries);
- Constraints (item defining the conditions that must be met in order to have a valid model);
- Relations (the syntax is similar to the syntax of programming language).

The knowledge records are composed of items. An item represents the smallest part of the knowledge record. The knowledge items are represented in the EBNF[1] (Extended Backus-Naur Form) notation.

The structure of the knowledge of the Extended CAD model is defined to be compliant with STEP (Standard for the Exchange of Product model data) ISO 10303 product structure [4]. Actually, the knowledge is defined and structured similarly to the structure of the product (Knowledge as the Product) and can be attached to the product definition as a definition of the product document (Document as the Product).

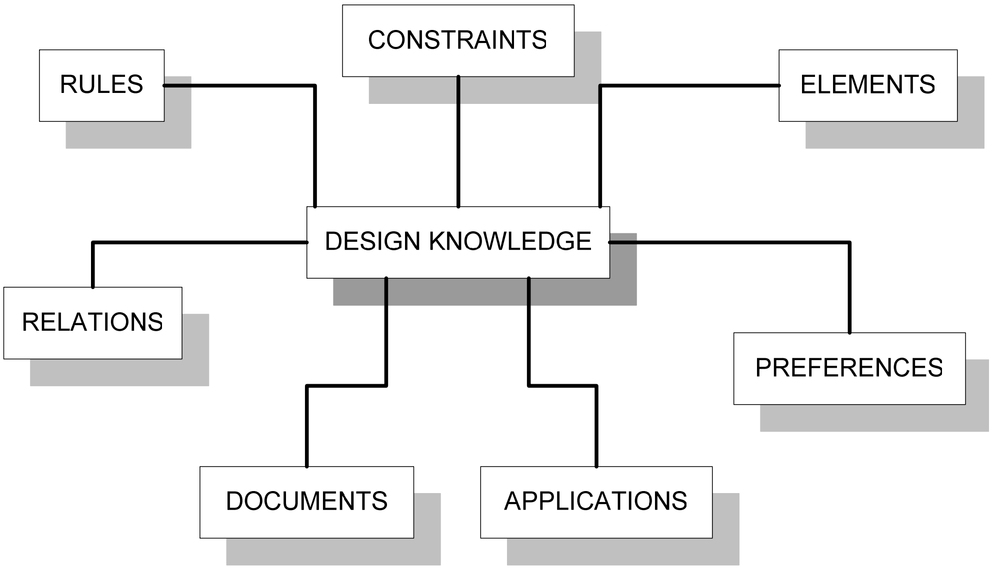

Figure 2. The structure of the Design Knowledge

3. The proposed model of design knowledge

The information model was created based on the proposed structure of the design knowledge. Several standard information models [5], [6], [7] were considered as the basis for the creation of the information model of design knowledge. Standard informational models were evaluated based on the requirements that were recognized as significant for a chosen model to meet. These requirements are as follows:

- Implementation using available CAD program applications;
- Support for geometric data manipulation;
- Ability to extend the model with non geometric data.

A couple of standard information models fulfill particular requirements, but the STEP [5] standard was found to be the most suitable for the task.

The Product entity in the STEP (AP 203) standard is considered to be the carrier of information. Therefore, we try to connect our model of design knowledge seamlessly to the existing entities of the STEP Product structure. Analyzing the STEP standard, the implementation of the *Document as the Product* was selected as the most suitable part to be altered and revised to implement design knowledge. In order to be able to implement the proposed structure of the design knowledge using Document as Product structure, several

[1] brief explanation can be fond at http://www.garshol.priv.no/download/text/bnf.html

alterations of the structure had to be done. The range of the attribute values of the entities from the Document as Product structure were extended with values required for the design knowledge implementation.

```
GLOBAL
  JEDINICE 1
  DOKUMENT 3
END GLOBAL
FOR verzija 1
  PROVJERI 1
  ELEMENT 2
  IF OGRANICENJE 3  THEN
      RELACIJA 1
      DOKUMENT 1
      DOKUMENT 2
      MATERIJAL 1
  ELSE
      RELACIJA 2
      DOKUMENT 1
  END IF
END FOR

FOR verzija2
  OGRANICENJE 2
  ELEMENT 3
  IF d20 > 9.0 THEN
      RELACIJA 2
      DOKUMENT 1
  END IF
  IF OGRANICENJE 9 THEN
      IF part18:d34 == 35 THEN
          RELACIJA 1
          MATERIJAL 2
      ELSE
          RELACIJA 2
      END IF
  END IF
END FOR
```

Figure 3. The Rule Code example

The most demanding task was to implement design knowledge element Rules (as part of design knowledge), because the Rules have a structure similar to the programming language (Figure 3). The number of lines (or items) is never final so this part of the structure should be able to expand or shrink dynamically. Figure 4 shows the structure of the proposed model of design knowledge.

The design knowledge is connected with the Product by using the *product_definition_formation* entity. The function of the Product is defined by the value of the name attribute of the *product_definition_context*. When this value is set to *design_knowledge* then the represented structure describes the design knowledge. Items that describe the design knowledge are defined using the *product_related_product_category* entity. The actual BNF data of a particular design knowledge item is represented by the *description_representation_item* entity. In addition to the design knowledge, operational and procedural knowledge are also represented using STEP.

A search for the right implementation platform has turned out to be a very demanding job. The problem was the customization of the CAD modeler to meet the requirements for the ECAD model implementation. This is due to the fact that professional development tools for the customization of CAD modelers costs a substantial amount of money (our project had limited resources). Therefore, the implementation had to be done using script languages available at the time. That is why, for implementation purposes, the following additional entities had to be created:

- Person and organization – holds the data about users (name, role, …) that use or create an extended CAD model;

- Knowledge about knowledge – holds the information about whereabouts of the created design knowledge on the computers system and how to access that knowledge;
- Security – holds the information about the design knowledge availability for viewing or modification by a particular user;
- Approval – holds the information about the design knowledge status and ability to be used in the actual design process.

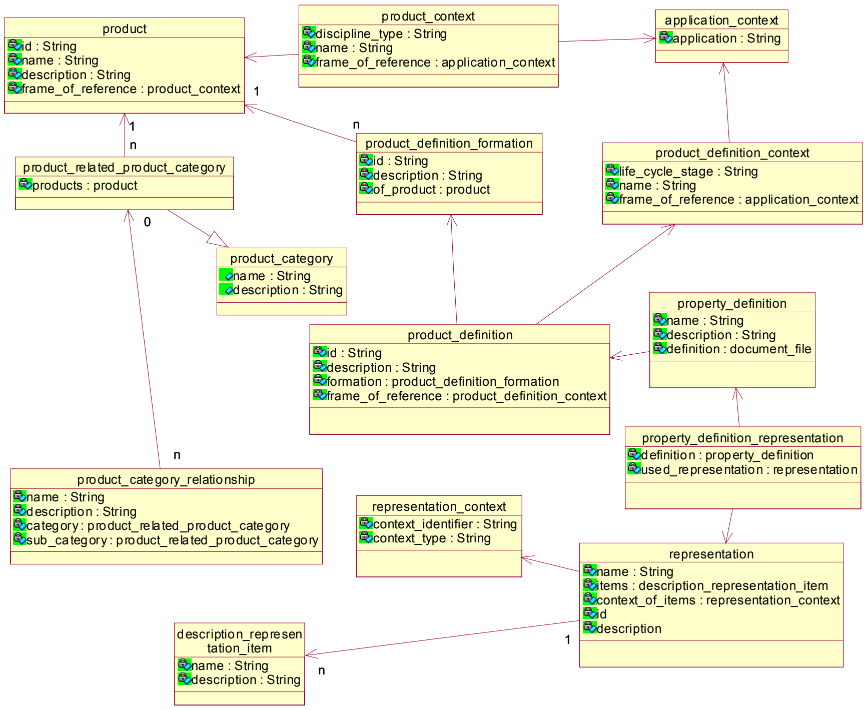

Figure 4. The proposed model of design knowledge (UML model [8])

In the process of creating the model of the design knowledge, the main considerations were that the proposed model should be compatible with the STEP standard. The compatibility is crucial because of the possibility of real implementation in different program systems that have to use or deal with design knowledge but do not use a graphic representation of the product. The compatibility also enables the exchange of the FBD model among different CAD applications without extra customization.

After the creation of all STEP [9] entities, the next step was to create a program system for the usage of the Extended CAD model. The first step was the creation of a relation database that would hold the design knowledge information. Several tables were created in the database with each table holding the data about a particular group of knowledge. The information from the database tables can be accessed from the program system in the usual manner (ODBC (Open DataBase Connectivity) or ADO (Active Data Objects)). The database structure of the proposed model of design knowledge is shown in Figure 5.

For the purpose of creating a program system, the object model (Figure 6) of the design knowledge model was created. The leading class is an abstract class Knowledge (Znanje). The classes that directly inherit properties from this class are the DesignKnowledge class (KonstrukcijskoZnanje) and the MetaKnowledge (ZnanjeZnanje) class. These classes are used for the representation of the defined design knowledge. Each object derived from a particular class has an associated record in a particular table in the database.

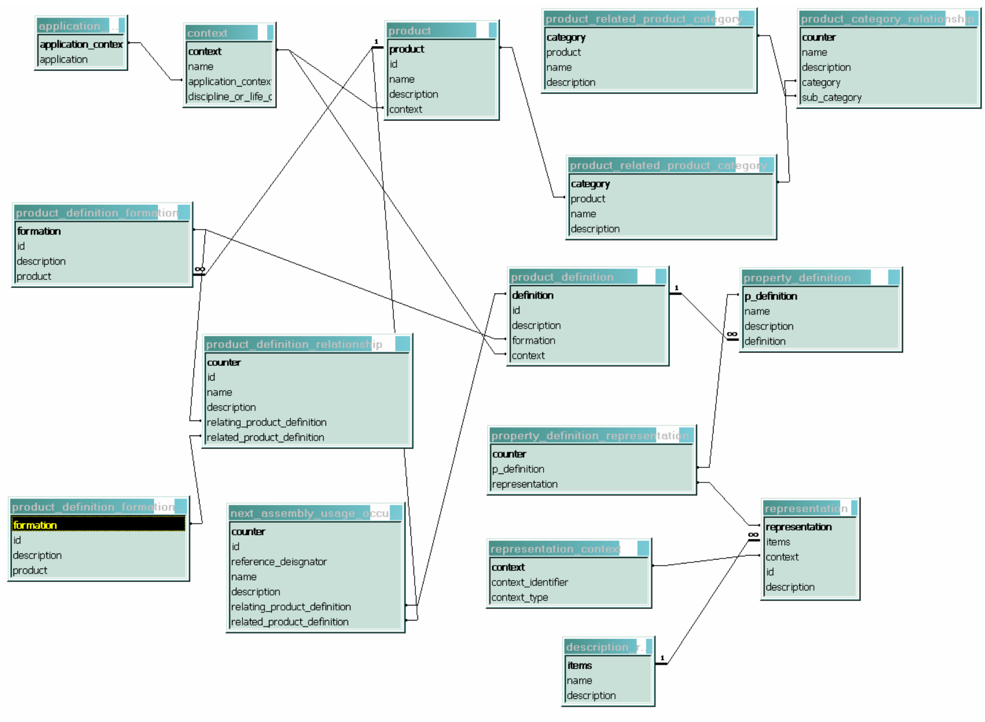

Figure 5. Database structure of the design knowledge model

During the program execution the persistence of the object data is achieved by using the method for dynamic binding. The method for dynamic binding loads the data from the database when a particular object is in scope and writes the data back to the database when the object is destroyed and the object data is changed. Dynamic binding is used due to the fact that during the usage of the program a large amount of the computer memory was used for the storage of data and this diminishes the program efficiency.

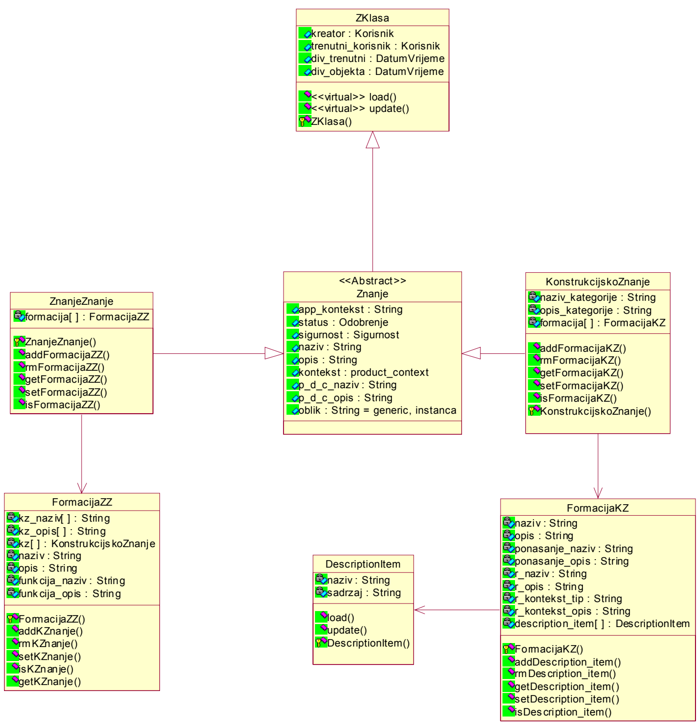

Figure 6. The structure of classes of the design knowledge model

4. Program system

The program system is created based on a defined class structure of the Extended CAD model objects [7]. The developed system consists of the following modules:

- Main module – creates and initializes objects, manages the program system; it also establishes a connection to the CAD application (if any) and manages the data transfer between the program system and the external application;
- Database interface module – creates a connection to the database, transfers and controls the data from the program system to the database;
- User interface module;
- Knowledge Interface module – handles various knowledge manipulation requirements (creation, modification or removal of the elements of the design knowledge);
- Auxiliary modules – are support modules for the interaction with libraries, for system initialization, path handling, awareness of critical situations during execution (CAD application crash, model or component removal, login failures, etc.).

The Extended CAD model is implemented in the JAVA programming language and tested in the Pro/Engineer environment.

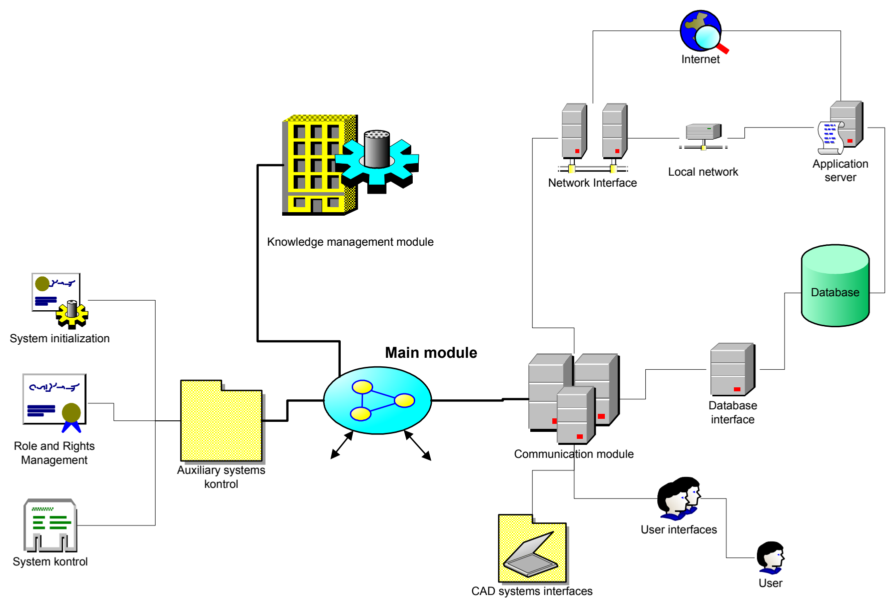

Figure 7. Extended CAD program system structure

The system was linked to the CAD modeler using J.Link libraries [10]. Through system interfaces (Figure 9), the relevant parameters, dimensions and features (that define a particular component version) are defined and initialized. During the creation of the product model, the user is able to choose among different versions of the Extended CAD model. The program system actively controls the regeneration of the model and its validity. The applied knowledge can be viewed or modified at any time online (by using the Pro/Engineer program system) or offline (by using MS Access Database). Strict control of the usage of ECAD is imposed by using security and roles for users.

The program system and the ECAD model were tested on a simple model of the power transformer product. The power transformer selected for testing consists of the following components:

- Core – three – coil version;
- Tank – polygonal version;
- Coolers – fan coolers on the tank long walls;
- Stiffeners – just on the side walls;
- Lamination steel – created as one part, not an assembly.

An example of the design knowledge captured during the work alongside with designers in the KPT company is given as follows:

- The position of the flanges for the power transformer core depends on the position of the core in the tank. The core position is determined by moving the centre of the core in the cross-section in relation to the centre of the tank in the cross-section. The flange positions in relation to the core have corresponded to the supports on the core itself. The longitudinal flanges are always facing the core centre, and the transversal flanges can be directed to (the outer) or from the centre (the inner). Flanges can be of various sizes.
- The core flanges my be, but do not need to be set. By the direction of the transversal (side) flanges, they are divided into outer flanges and inner flanges. The flange position is defined in relation to the local co-ordinate system representing the geometric origin of the core in the cross-section. The core origin is placed in relation to the geometric origin of the tank. A movement in the positive direction along both co-ordinate axes places the origin in the quadrant I. The maximum number of side flanges is three, and the minimum two. The position of central flanges is determined in relation to the transversal centerline of the core. The left and the right flange on the long wall are always the same, as well as the sides HV (High Voltage) and LV (Low Voltage). The values for the positions of the flanges on the long and side wall flanges have to correspond with the values of support positions near the transformer core. The translation of the transformer core in relation to the tank is separately defined.

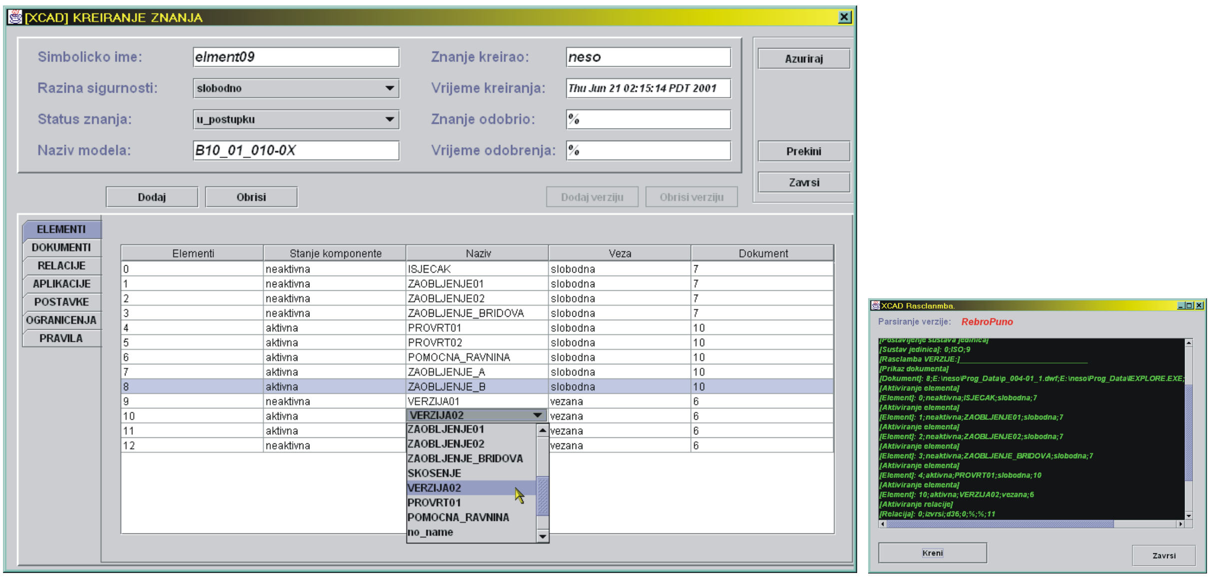

Figure 9. On the left: one of the dialogs for the creation
of elements of the design knowledge;
on the right: the dialog for insertion of the defined Extended
CAD model into the existing assembly

Using the defined set of programming instructions, several different versions of the product were defined (one is shown in the figure 10). Four assemblies were created in Pro/Engineer using various versions of Extended CAD models. The Extended CAD model was proven to be adequate for the reuse of existing design but not for the creation of a new design from scratch. The time spent on the design of the power transformer using the Extended CAD model has decreased by about 25%.

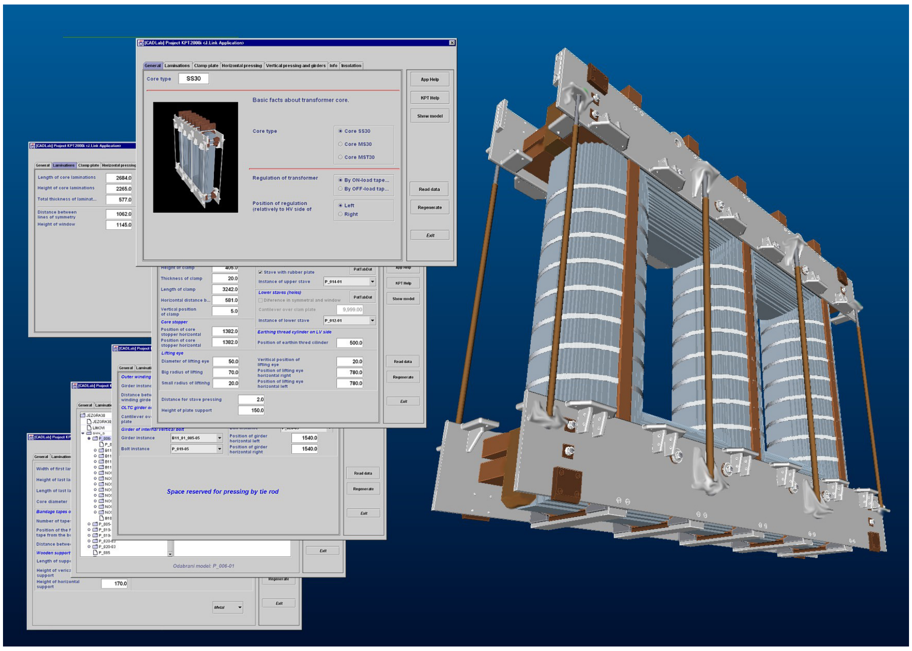

Figure 10. One version of the computer model of the power transformer core

5. Conclusion

The realized Extended CAD model exhibits quasi intelligent behavior represented through the following:

- recognition of the existing status of the Extended CAD model,
- interaction with the user during the creation and usage of the Extended CAD model,
- maintenance of the validity of the FBD CAD model,
- invocation of the use case scenario if the conditions of the model are fulfilled.

The realized Extended CAD model can be used as the basis for the creation of more advanced knowledge supported CAD systems. The next step will be to implement learning capabilities into the computer system so that the Extended CAD model can learn from the user and expand the existing knowledge.

6. References

[1] M. Bangemann, Europe and Global Information Society – Recommendations for the European Council, Brussels, 3. May 1994.

[2] M. Jovic, N. Bojcetic, Development of the Mechanical Assembly Description Model, Proceedings of the 5th International Design Conference Design 98, Dubrovnik, 1998.

[3] D. Marjanovic, N. Bojcetic, D. Dekovic, N. Pavkovic, D. Pavlic, M. Storga, D. Zezelj, Design Department Data Flow Integration, Integrated Product Development Proceedings of the 3rd International Workshop, Magdeburg, 2000.

[4] J. Flowler, STEP for Data Management, Exchange & Sharing, Technology Appraisals, Twickenham, 1995.

[5] Introducing STEP, http://strategis.ic.gc.ca/STEPguide

[6] IDEF1X Manual USAF Integrated Computer-Aided Manufacturing Program, D'Appleton Company, USA 1986.

[7] Deux, The Storz of O2, IEEE Trans. Knowledge and Data Engineering, Vol. 2, No 1, 1990.

[8] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, Reading, 1999.P. Gu, K. Chan, Product modeling using STEP, Computer-Aided Design, pp. 163 – 181, Vol. 27, No. 3., Butterworth-Heinemann, Oxford, 1995.

[9] K. H. Law, T. Barslow, G. Widerhold, Management of Complex Structural Objects in a Relational Framework, Engineering with Computers, pp. 81-92, Vol. 6, Springer-Verlag, New York, 1990.

[10] J-Link User's Guide, PTC - Documentation, Waltham, 2000.

Dr. sc. Nenad Bojcetic
Faculty of Mechanical Engineering and Naval Architecture
Chair of Design
Tel.: +38516168121, Fax.: +38516156940, email: nenad.bojcetic@fsb.hr
url: www.cadlab.fsb.hr
Ivana Lucica 5, 1000 Zagreb
CROATIA