Modeling Knowledge for Knowledge-based Design Support

P.M. Wognum N.J.I. Mars
University of Twente
Department of Computer Science
Knowledge-based Systems Group
Enschede

1. ABSTRACT

One of the research projects of the Knowledge-based Systems Group of the University of Twente is aimed at developing models and techniques to support engineering design in technical domains like mechanical, electrical, and civil engineering. Our goal is to offer knowledge-based support during all phases of the design process, from initial requirements to a design description which can easily be manufactured.

Engineering design is a knowledge-intensive process requiring many different types of knowledge from rather diverse sources. To be able to develop intelligent support systems which manage the knowledge in a shareable and integrated fashion, we need a model of the design process. We present an outline of such a model which integrates several process aspects.

To create shareable, re-usable knowledge bases which can be used in an integrated way we also need ontologies to structure the knowledge acquired for the design task. We present an ontology for structuring product-related knowledge which enables us to integrate knowledge from several engineering design disciplines.

The ontology can be used to analyze existing bodies of knowledge, like standards, cases, and physical effects. The knowledge thus structured can be integrated in the design process. We present our research projects which focus on modeling existing knowledge by means of the product ontology presented before.

2. INTRODUCTION

One of the research projects of the Knowledge-based Systems Group of the University of Twente is aimed at developing models and techniques to support engineering design in technical domains like mechanical, electrical, and civil engineering. Our goal is to offer knowledge-based support during all phases of the design process, from initial requirements to a design description which can easily be manufactured.

An important requirement for the support systems we envision is that the systems must be shareable, reusable and must be integrated with other tools and systems. To enable sharing and reuse we must develop shareable ontologies which incorporate generally accepted concepts in the design domain (Neches et al. [1991]). In the same vein, to facilitate integration, we need languages which act as intermediates between several implementation platforms (Cutkosky et al. [1993], Alberts, Wognum & Mars [1993]). Another requirement is that the support systems must be modular and flexible in the sense that they must be easily adaptable to changing situations.

Design, however, is a knowledge-intensive process requiring the use of different types of knowledge coming from rather diverse sources. The knowledge used during the design process also serves different purposes, such as the transformation of product descriptions, the evaluation of those descriptions against many different requirements, including analyses like kinematics, tolerances, and assemblability, but also the selection of suitable materials. Moreover, knowledge is needed about the required content of design description in different phases of the design process to adequately perform the evaluations. Performance criteria on the design process are needed to assess the results of these evaluations. These criteria may concern the time spent in the design process, quality of the designed product, and the costs involved.

To be able to use the knowledge intelligently during the design process, and to flexibly adapt to changing situations, we must understand the process thoroughly. In particular, we must understand what knowledge is used in which phase of the design process and for what purpose. To this end, we need a process model which captures different aspects of a design process. We are currently developing such a model. The model will be briefly described in section 3.

To enable the shared and integrated use of the knowledge during the design process, we need models, also called ontologies, for structuring the different types of knowledge. These ontologies must consist of generally accepted concepts and relations between them. Such ontologies help in analyzing and structuring the knowledge to be acquired for use in a design support system. In our project we have developed an ontology for analyzing and structuring product-related design knowledge, a very important part of all design knowledge. The product-related knowledge provides the basis for design descriptions in a specific design process and for the transformations of these descriptions from the initial specification into the final description of the product to be designed. The ontology is currently restricted to the physical technical part of design descriptions which can be expressed in mathematical terms.

The ontology can be used to analyze and structure product-related knowledge in different engineering domains thus enabling sharing and integration. In addition, the ontology integrates two important aspects in one design description, namely behavior and form, which consists of material and geometrical properties. The integration of these two aspects enables the integrated use of knowledge related on the one hand to drawing and to analyses like strength computation on the other hand. The ontology will be described in section 4.

The product knowledge ontology is also used in our project to enable the analysis of large bodies of knowledge which can be used in specific tasks, such as configuration, evaluation, proposing a design, redesign, or solving a difficult technical conflict. The knowledge needed to perform these different tasks consists of knowledge of a specific product class, product standards, cases, redesign knowledge, and physical knowledge like effects, respectively. We will describe the different analyses in section 5.

3. PROCESS MODEL

In our view, the design process is in essence an information-processing process. At the start of the process the information consists of the specifications of the product to be designed. At the end the information consists of the description of the artifact ready to be manufactured.

In modeling the design process we need to take into account several aspects. First, we must model the main flow of information consisting of descriptions of the product to be designed in each phase of the design process. Second, we must model the control flow of information which consists of the status of the design description at particular stages of the design process. The control flow models the sequence of transformations on the design description. Third, we must model the support needed in each transformation on the design description in terms of the knowledge needed for each such process.

Most current modeling techniques, however, focus on isolated aspects of the design process. For instance, data flow diagrams focus on the information flow through the process. The resulting description is a static one, ignoring the dynamic aspects of the process. Similarly, petri nets, as often used in workflow analysis and design, focus on the control flow of a process. In the resulting description the content of the information is often ignored. The knowledge and data needed to perform the processes, often absent in both data flow diagrams and petri nets, require separate modeling techniques such as entity-relationship diagrams.

In modeling design processes, these different aspects need to be integrated to allow us to develop support systems in which the different types of knowledge needed during the design process can be explicitly recognize and managed. In addition, such support systems can better be adapted to changing circumstances. For example, if design process procedures in an organization change, it can be expected that the systems to support the design process must be adapted to reflect those changes. In such a case the effect of the procedural changes, which mainly concern the control aspect, must be propagated to the knowledge needed to perform the process and the consequences for the design description in the design process phases involved.

We have recently started the development of a model of the design process which explicitly models the dependencies between the different aspects described above. The process model is based on General System Theory (Wognum [1995]). We test and refine the model by means of case studies in industry. The case studies are performed with the help of Master students in cooperation with the School of Management Studies at our university.

The case studies are part of the ConcERT programme (Boer [1994]). The goal of this research programme is to develop a method to support industries in adopting and implementing Concurrent Engineering in their organization. Concurrent Engineering (CE) in this context is viewed as an approach to design in which both the product and the processes acting on the product are designed in an integrated way. CE is aimed at reducing the throughput time of the design and production process, reducing the costs and enhancing the quality of the end product. Implementing CE requires both organizational and technological changes in the industries involved.

The model introduced above will enable us to more formally define the different knowledge types needed during the different phases of the design process including knowledge to control the process. It will also support us in acquiring the knowledge and in developing new ontologies or extending existing ones for analyzing and structuring the knowledge. Finally, it will assist us in choosing languages to represent the knowledge in such a way that it can easily be retrieved and used for (automated) reasoning.

In addition, the model will assist us in developing suitable design support systems for specific design processes. It will offer a basis for developing techniques for managing the knowledge needed during the design process in an integrated way.

An important part of the knowledge used during the design process consists of knowledge from which the product description is composed. In the next section, an ontology is presented which models the technical part of this product-related knowledge.

4. PRODUCT MODEL

In modeling the knowledge from which the product is to be composed, again several aspect need to be taken into account. Most importantly, the function of the design must be modeled, the behavior which implements the function, and the form aspects, like the geometrical and material properties of the design which determine its behavior.

In most existing approaches these aspects are often modeled separately. As a result, special evaluation procedures are needed to integrate the different aspects explicitly. By integrating the behavior and form aspects into one model, such 'after the fact' evaluations are no longer necessary.

We have developed an ontology for structuring engineering design knowledge, called YMIR (Alberts [1993]), which integrates behavior and form. The YMIR ontology specifies a taxonomy of concepts for engineering design which define the semantics of design knowledge in multiple engineering domains. The concepts in YMIR represent generalisations of the concepts used in the individual design domains, such as electrical, mechanical, and civil engineering. The more specific concepts in each domain are instantiations of the general YMIR concepts.

In the design process, the product knowledge is used to transform and refine the initial specifications into a detailed description of the final artifact. In essence, the design process can be regarded as the problem of finding a configuration of physical elements in a single artifact that can perform a particular function as specified in the initial specifications. These physical elements have particular geometrical and material properties, called the form, and display a certain behavior that is dependent of the form. A function is the required part of the combined behavior of a configuration of elements, which means that specific values have been found for the forms of those elements. These values should be in accordance with the requirements on the geometrical shape and materials of the final artifact.

The concepts defined in the ontology specify generalized physical elements in an engineering domain which can be used as building blocks from which a design description is composed. There is, however, a large gap between the initial specifications and the final description of the artifact to be designed. To bridge this gap, several intermediate levels of abstraction are required. At each level, a separate set of building blocks has to be defined. A design description composed of building blocks at one level serves as a specification for a design to be found at the next lower level of abstraction. This way, engineering design can be modeled as a process which allows us to systematically describe all design phases from conceptual design to detailed design in terms of combining basic 'building blocks'.

Provided high-level abstract building blocks can be specified for the initial design phases, there is no fundamental difference between conceptual design and detailed design except for the level of abstraction of the building blocks used. In VLSI-design, for instance, this approach is already common practice (Alberts [1993]).

We have defined a type of generic building blocks based on system models in System Theory, called generic system models (GSMs), for synthesising product descriptions. These generic system models constitute the bottom-line of the taxonomy of concepts defined by the YMIR ontology. The main arguments for building on a well-founded theory such as System Theory are that it allows for the formal verification of the results obtained, and facilitates the integration of existing engineering techniques. The particular type of system models we are interested in is known as network models (MacFarlane [1970]). We have chosen for this particular class because of its importance as a widely accepted representation for describing technical systems in many engineering domains.

The ontology can be shared by different design disciplines involved in the design process. Examples of the use of the ontology in the area of bridge design can be found elsewhere (Alberts & Dikker [1994]).

We are currenty refining the abstraction operators for translating a design description from one abstraction layer into another. This research is performed in a SION project called 'Model-based abstraction in engineering design'. In addition, we refine our generic system models to explicitly integrate multiple engineering disciplines.

5. EXAMPLES OF KNOWLEDGE ANALYSIS

The ontology introduced in the previous section has been used to study and model existing bodies of design knowledge. In the next sections, we will present several research projects in which YMIR has been used to analyse such bodies of knowledge.

5.1 Oil compressors

In this subproject, a case study is performed to test YMIR on real product knowledge. The knowledge consists of configurations of oil compressors (van Raalte [1993]). The questions to be answered by this project are;

- Is YMIR suited to model a real product?
- What part of the knowledge can not (yet) be modeled with YMIR?

Currently, a shell is being developed to support the representation of and reasoning with the product knowledge. Students of the School of Polytechnics in Enschede have supported part of this work (Bom & Bouma [1994]).

After finishing the shell, we will develop the compressor knowledge base in cooperation with Delaval Stork in Hengelo. At that time, we will be able to answer the questions raised above. In addition, several other examples will be worked out.

5.2 **Product standards**

In a design process, many requirements play a role in addition to the functional specifications. These requirements are often left implicit in the specification phase of the design process. They are incorporated in the design process as soon as they become relevant to the design at hand. Examples of such requirements are design codes and safety requirements and are rather general in nature, which means that they are applicable to a large variety of product designs.

In this part of our project, we investigate how requirements can be modelled on the basis of YMIR and how they can be used in the design process as early as possible. Our objective is to apply requirements to the design description as soon as they become relevant to the design at hand (Dikker & Wognum [1993a], Dikker & Wognum [1993b], Dikker, Alberts & Wognum [1993]). As a result, conflicts are detected as early as possible thus preventing unnecessary iterations.

To be able to formally structure requirements, we have developed a model of requirements (Dikker et al. [1994]) which consists of a scope part and a restriction part. The scope part is a design description representing the situations in which the requirement is applicable. The restriction part describes the actual requirement and restricts specific parts of the scope. With the help of this requirement model, requirements can be used in the design process as desired. The scope part can be matched with the description of the design under construction to determine whether the requirement is applicable. When after matching an applicable requirement is found, the restriction part can be applied to the design at hand possibly leading to the elimination of possible instantiations. If, as a result, the design description becomes inconsistent, resolution of the conflict is needed. This part requires knowledge to control the design process and is left for future research.

The approach has been implemented using Galileo (Bowen [1992]), a constraint programming language. Currently, the requirement model is tested against a number of design codes. In june 1995, the results of this study will be published in a Ph.D. thesis.

5.3 **Cases**

In practice, designers often reuse earlier experience in new design situations. Such experience mainly exists in the form of already finished (successful or possibly rejected) design cases. The main problem in reusing such knowledge is in formalizing, representing, retrieving and adapting old cases to satisfy new requirements.

In this project, we study how existing cases can be retrieved to be used as starting points for a new design problem. Such a problem consists of new functional specifications for (part of) a product to be designed at any point during a design process. To be able to reuse cases, we need methods and techniques for selecting them. We further assume that cases are represented as YMIR product models.

A case is considered to be 'most suitable' if it requires the least adaptation compared to other candidate cases. To be able to determine the suitability of a case, we first need to classify adaptation according to its complexity in terms of, for instance, the number of computation steps involved in performing the adaptation. Second, we need models and techniques to designate the case which best suits the new specifications for (part of) the design at hand.

To predict the possible adaptations needed for a case, we may use traditional case-based reasoning techniques extended with knowledge on the complexity of adapting cases. We should aim at restricting the additional amount of knowledge as much as possible, however, since one of the advantages of case-based reasoning is deemed to be the avoidance of the knowledge acquisition bottleneck. The addition of a large amount of adaptation knowledge clearly removes this advantage.

It is expected that case-based design enhances the efficiency of the design process. The objectives of this subproject are to define indices to enable retrieval of cases which are candidates for a new design problem, to model adaptation knowledge to determine similarity of the candidate cases with the new design problem, and to test the model and techniques in several practical situations. The results of this project will be published in a Ph.D. thesis in October 1996.

5.4 **Redesign knowledge**

This subproject of our design project concerns the process of redesigning a specific technical system to let it meet new requirements. The project is performed in the context of the SION special attention area 'Evolutionary knowledge-based systems', called Revise. This attention area is a cooperation between the UT, UT/ECN, UvA, and VU.

The aim of redesign, in general, is to produce a new design in a very efficient way. The possibility to design a new product by slightly modifying an old design, often results in a reduction of costs and effort in the design process. The success of redesign depends on the similarity of an old design with the new one. When the modifications that have to be made, are large, redesign is not a very good idea; designing from scratch seems then more appropriate.

For our subproject we assume that an old design description is given. The process of selecting a relevant case from a case-base is the subject of the subproject described above. Besides an old design, we assume new specifications, which differ from the specifications of the old design. Further, both design and specifications are described in the same way, within our modeling framework YMIR.

The output of the redesign process is a new design, which meets the new specifications. To accomplish this result, several subtasks can be distinguished:

- Diagnosis: The localization of the part of the original design which has to be adapted and the part of the design which may remain the same.
- Respecification: For the part which has to be adapted, a new specification has to be found, such that the overall new specification is met.
- Design: This partial specification must actually be designed. Several tasks are possible, like design from scratch, case-based design, redesign again, or enlisting a human designer.
- Evaluate: The result of the design step has to be evaluated. It has to be checked whether the new overall design meets the required overall specification.

The crux of redesign is formed by the diagnosis and respecification subtasks. At the moment we study Model-Based Diagnosis techniques for accomplishing these two tasks. To this end, we consider a design description as a MBD-model, consisting of components and connections and functional descriptions of the components. The difference between a new specification and an old design can be seen as the equivalent of a MBD conflict. The result of our use of MBD is a new partial specification, the equivalent of a MBD diagnosis. For details, we refer to (Bakker et.al. [1994]).

We are about to start with the analysis of redesign knowledge with the goal to identify different classes of complexity in the redesign process. The diagnosis and respecification phase are expected to be influenced by the complexity level of the redesign process (Eldonk et al. [1995]).

5.5 **Physical knowledge**

In the fifties, after G. Altshuller had studied a large number of patents from various patent collections, he concluded that most innovative design solutions were obtained by the transfer of ideas from one engineering domain to another. By "idea transfer" Altshuller means the transfer of a known physical or design principle (conceptual design knowledge, in other words) from a domain in which the principle has been effectively used to another domain where the use of this principle was unknown or its applicability has not been investigated yet.

Altshuller further developed this idea and formalized it in TIPS, the Theory of Inventive Problem Solving. The theory was tested thoroughly and proved the workability of Altshuller's concept. For more details on TIPS, we refer to (Sushkov, Mars & Wognum [1995]).

One important characteristic of an innovative design situation as found by Altshuller is a conflict in the current design which cannot be solved by optimizing the parameters in the design. To eliminate the conflict, the current physical structure of the design must be changed. TIPS provides a structured process of using general principles of innovative design to find conceptual knowledge in another engineering domain or in physics. This knowledge can be instantiated into a new structure for the design.

To organize innovative design as a process which reuses conceptual knowledge, as suggested by TIPS, we need to represent this knowledge in a form which will make it possible to share this knowledge between different engineering domains. Examples of such abstract knowledge are physical effects which can be instantiated into numerous technical solutions.

We currently focus on the identification and classification of the different types of functions which have to be achieved by a design in a specific context. In addition, we try to model physical effects with the help of YMIR which can help to implement the desired functions.

6. **SUMMARY**

We have presented our project on developing models and techniques for supporting engineering design. We have briefly discussed the need for a process model which allows us to capture the different aspects of a design process, and, more specifically, identify the different types of knowledge needed in the design process. The dependencies between the different aspects, such as information flow, control flow, and the means to transform information during the process, are modeled explicitly. This way, changes in one aspects can be propagated into the other aspects.

We have also discussed an ontology for structuring an important part of the knowledge, namely the technical part of the product-related knowledge. The ontology specifies a taxonomy of concepts with a well-defined semantics in several engineering domains. The ontology integrates two important aspects of a product description. These aspects are the behavior of a design and the form, consisting of material and geometrical properties which determine the behavior.

The ontology is used to model existing bodies of knowledge, such as product norms, standards, cases, redesign knowledge, and abstract physical effects. By basing the structuring of this knowledge on the same underlying concepts, the knowledge can be shared and used in an integrated way in the design process.

7. ACKNOWLEDGEMENTS

We thank the members of our group who have contributed to the work reported in this paper. These members are Mert Alberts, Frank Dikker, Frank Tempelman, Bas van Eldonk, Valeri Sushkov and Frank van Raalte.

8. PUBLICATIONS

L.K. Alberts[1993], “Ymir: an ontology for engineering design,” University of Twente, Ph.D. Thesis, Enschede, The Netherlands.

L.K. Alberts, ed.[1994], “Workshop notes of the AID94 workshop on semantic basis for sharing of knowledge and data in design,” Lausanne, Switzerland.

L.K. Alberts & F. Dikker[1994], “Integrating standards and synthesis knowledge using the Ymir ontology,” in Proceedings of the Third International Conference on Artificial Intelligence in Design, 15-18 August 1994, Lausanne, Switzerland, J.S. Gero, ed., University of Twente, Enschede, The Netherlands.

L.K. Alberts, P.M. Wognum & N.J.I. Mars[1993], “Integration of design knowledge into step product models,” in Proceedings of the 9th International Conference on Engineering Design, N.F.M. Roozenburg, ed., Heurista, Zürich, Switzerland, 1627--1633.

R.R. Bakker, S.J.M. van Eldonk, P.M. Wognum & N.J.I. Mars [1994], “The use of model-based diagnosis in redesign,” in Proceedings of the 11th European Conference on Artificial Intelligence, A.G. Cohn, ed., John Wiley & Sons, Chichester.

A.J.B.A. Bom & F.R. Bouma[1994], “Ontwerp en implementatie van een interface tussen Ymir en AutoCAD,” Universiteit Twente, Afstudeeropdracht HIO-Enschede, UT-KBS-94-17, Enschede.

J.A. Bowen & D. Bahler[1992], “Frames, quantification, perspectives, and negotiation in constraint networks for life-cycle engineering ,” Artificial Intelligence in Engineering7, 199--226.

M.R. Cutkosky, R.S. Englemore, R.E. Fikes, M.R. Genesereth, Th.R. Gruber, W. Mark, J.M. Tenenbaum & J.C. Weber [1993]], “PACT: an experiment in integrating concurrent engineering systems,” IEEE Computer.

F. Dikker, L.K. Alberts & P.M. Wognum[1993], “A constraint-based approach to design critiquing,” in Proceedings of the 6th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, Edinburgh, UK, P.W.H. Chung, G. Lovegrove & M. Ali, eds., Gordon and Breach Science Publishers, Yverdon, Switzerland, 6--9.

F. Dikker, L.K. Alberts, P.M. Wognum & N.J.I. Mars[1994], “Structuring design requirements using Ymir,” University of Twente, Memorandum Informatica 94-41, Enschede, The Netherlands.

F. Dikker & P.M. Wognum[1993a], “Constraint-based critiquing: a knowledge-based approach to design for quality,” in Proceedings of the 9the International Conference on Engineering Design, The Hague, N.F.M. Roozenburg, ed., Heurista, Zürich, Switzerland, 926--933.

F. Dikker & P.M. Wognum[1993b], “Knowledge-based quality assurance: a critiquing approach to support ISO9000 in design,” University of Twente, Technical Report UT-KBS-93-22, Enschede, The Netherlands.

S.J.M. van Eldonk[1994], “Literature report on redesign,” Technical Report, UT-KBS-94-22, Enschede, The Netherlands.

S.J.M. van Eldonk, L.K. Alberts, R.R. Bakker, F. Dikker & P.M. Wognum[1995], “The redesign of technical systems,” Knowledge Based Systems, To appear.

A.G.J. MacFarlane[1970], Dynamic system models, George G. Harrap & Co., Ltd., London.

R. Neches, R. Fikes, T. Finin, Th.R. Gruber, R. Patil, T. Senator & W.R. Swartout[1991], “Enabling technology for knowledge sharing,” AI Magazine12, 36--56.

V.V. Sushkov[1994], “Reuse of physical knowledge in creative engineering design,” in Workshop Notes of the ECAI94 Workshop on models and techniques for reuse of designs, P.M. Wognum, ed..

V.V. Sushkov, N.J.I. Mars & P.M. Wognum[1995], “Introduction to TIPS: a theory for creative design,” AI in Engineering, to appear.

F. Tempelman, P.M. Wognum & N.J.I. Mars[1994], "Case-based reasoning in literature and Stevin," University of Twente, Technical Report UT-KBS-94-12, Enschede, The Netherlands.

P.M. Wognum, ed.[1994], "Workshop Notes of the ECAI94 Workshop on models and techniques for reuse of designs," Amsterdam.

P.M. Wognum[1995], "A system-theoretical view to knowledge-based systems," University of Twente, Technical report UT-KBS-95-10, Enschede, The Netherlands, in preparation.