knowledge engineering in computer-aided design

edited by
j.s. gero

IFIP

north-holland

KNOWLEDGE ENGINEERING IN COMPUTER-AIDED DESIGN

Proceedings of the IFIP WG 5.2 Working Conference on
Knowledge Engineering in Computer-Aided Design
Budapest, Hungary, 17-19 September, 1984

Edited by

John S. GERO
Department of Architectural Science
University of Sydney
Sydney
Australia

1985

NORTH-HOLLAND
AMSTERDAM · NEW YORK · OXFORD

ISBN: 0 444 87825 4

Published by:
ELSEVIER SCIENCE PUBLISHERS B.V.
P.O. Box 1991
1000 BZ Amsterdam
The Netherlands

Sole distributors for the U.S.A. and Canada:
ELSEVIER SCIENCE PUBLISHING COMPANY, INC.
52 Vanderbilt Avenue
New York, N.Y. 10017
U.S.A.

Library of Congress Cataloging-in-Publication Data

IFIP WG 5.2 Working Conference on Knowledge Engineering in Computer-Aided Design (1984 : Budapest, Hungary)
Knowledge engineering in computer-aided design.

Bibliography: p.
Includes index.
1. Artificial intelligence--Data processing--Congresses. 2. Computer aided-design--Congresses. 3. Engineering--Data processing--Congresses. I. Gero, John S. II. Title.
Q336.I36 1984 620'.00425'028563 85-15876
ISBN 0-444-87825-4

PRINTED IN THE NETHERLANDS

IFIP WG 5.2 Working Conference on
Knowledge Engineering in Computer-Aided Design
Budapest, Hungary, 17-19 September, 1984

Conference Chairman:
J. S. Gero

International Program Committee:
J. S. Gero (Chairman), T. Vamos (Co-Chairman), A. Markus (Secretary),
A. Bijl, H. Brown, C. Eastman, J. Encarnaçao, S. Fenves, P. Hajek,
J. Hatvany, T. Havranek, J. Lansdown, V. Lane, J.-C. Latombe, L. Lopez,
S. Ohsuga, K. Preiss, A. Radford, E. Sandewall, E. Schlechtendahl,
E. Tyugu, E. Warman, D. Willey, and H. Yoshikawa

NORTH-HOLLAND
AMSTERDAM · NEW YORK · OXFORD

PREFACE

IFIP's Working Group 5.2 on Computer-Aided Design very early recognised the important role that artificial intelligence was likely to play in computer-aided design. Its first conference on this topic was held in France in 1978 (Latombe, 1978). Since then considerable formalisation and stratification has taken place in artificial intelligence. A subset of artificial intelligence labelled "knowledge engineering" has evolved. It has been elucidated by Feigenbaum (1977), who defined the activity of knowledge engineering as follows:

> "The knowledge engineer practices the art of bringing the principles and tools of artificial intelligence research to bear on difficult application problems requiring experts' knowledge for their solution. The technical issues of acquiring this knowledge, representing it, and using it appropriately to construct and explain lines of reasoning are important in the design of knowledge-based systems ... The art of constructing intelligent agents is both part of and an extension of the programming art. It is the art of building complex computer programs that represent and reason with knowledge of the world."

The fundamental structure used to represent reasoning and, hence, knowledge, is symbolic inference. The obvious advantage of inferencing is that it does not require an a priori mathematical theory; it can be used to manipulate concepts. Barr and Feigenbaum (1981), talking about the applicability of knowledge engineering in conceptual areas, state:

> "Since there are no mathematical cores to structure the calculational use of the computer, such areas will inevitably be served by symbolic models and symbolic inference techniques."

The industrial revolution saw the automation of mechanical power. The introduction of computers saw the automation of calculation. Now, knowledge engineering brings the automation of reasoning and with it the extension of the applicability of computers to both non-numeric and non-algorithmic computation. Since much of design knowledge falls into these two areas knowledge engineering has the potential to be of significance in computer-aided design. It is likely to change computer-aided design because it is now possible explicitly to include not just causal knowledge but also phenomenological and experiential knowledge in the computation process.

The aim of this working conference was to provide a forum for the exchange of ideas and experiences related to knowledge engineering in computer-aided design, to present and explore the state-of-the-art of knowledge engineering in computer-aided design, to promote further development, and to delineate further directions for research and application.

Sixteen papers from eight different countries on various aspects of the state-of-the-art in knowledge engineering in computer-aided design were presented. These covered the range from concepts through tools, from general applications to specific applications.

The fields of application included architecture, building, computer engineering, computer science, mechanical engineering, structural engineering and VLSI design. The presentations were followed by extensive discussions from the invited attendees. The edited discussions form an integral part of this volume.

There is an implicit structure in the ordering of the papers. The first two, both from Japan, are concerned with basic ideas of using knowledge in CAD systems. The next three, one from Australia and two from the USA, are all concerned with applying knowledge engineering within the domain of building design at different levels of granularity of knowledge. The next three, from the UK, Ireland and France respectively, take a step back from domain-specific applications and examine some general modelling issues. The next three, from Romania, USA and UK respectively, are all concerned with different aspects of expert systems. The next four, one from the USA, one from the UK and two from France, are all concerned with various aspects of applying knowledge engineering in the computing domain, for both hardware and software design. The final paper, from Hungary, takes a somewhat wider ambit and looks at knowledge and design and their places in flexible manufacturing systems.

Whilst the application papers traverse across many domains the ideas and concepts that are used appear to be widely applicable well outside the specific application domain. Such papers can, therefore, be read at both levels: elaboration of general knowledge-based ideas and exemplars of domain specific applications.

The success of the conference was not only due to the assiduous activities of the international program committee and the speakers and attendees (who came from 14 countries) but to the unflagging energy of Andras Markus, the local secretary. The discussion bears the unmistakable marks of Fay Sudweeks, my secretarial assistant, who single-handedly transcribed the entire discussion (and Dr Hatvany's presentation) from the tapes, with their inevitable static, and who helped me convert them to a semblance of English. She also looked after all the drafts of the discussion and its final presentation - special thanks are due to her.

John Gero
Sydney University
1985

Barr, A. and Feigenbaum, E.A. (1981). Handbook of Artificial Intelligence, Vol.1, William Kaufmann, Los Altos.

Feigenbaum, E.A. (1977). The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering. IJCAI-77, William Kaufmann, Los Altos, pp.1014-1029.

Latombe, J -C. (1978). Artificial Intelligence and Pattern Recognition in Computer Aided Design. North-Holland, Amsterdam.

TABLE OF CONTENTS

Knowledge Engineering in Computer-Aided Design
J.S. Gero (Editor)
Elsevier Science Publishers B.V. (North-Holland)

REQUIREMENTS AND PRINCIPLES FOR INTELLIGENT CAD SYSTEMS

Tetsuo TOMIYAMA
Hiroyuki YOSHIKAWA

Department of Precision Machinery Engineering
Faculty of Engineering
the University of Tokyo
Hongo 7-3-1, Bunkyo, Tokyo 113, JAPAN

The main aim of introducing CAD systems is to increase designers' creativities. However, it is doubtful whether designers are really supported by such CAD systems, for example, in conceptual design. Problems of conventional CAD systems are discussed in this paper, and we study the feasibilities of future CAD systems that can really help designers very intelligently and efficiently. An experimental knowledge based system is developed to show that those future CAD systems will be realized by introducing knowledge engineering.

1. INTRODUCTION

Conventional CAD (Computer Aided Design) systems have been contributing to rationalizations of the machine design process in drawing, checking the motions of a mechanism, generating NC machining data, or generating geometrical data for characteristics analyses, e.g., FEM. This means, computers made designers free from manual work of machine design by giving them time to devote themselves more deeply to the products. We agree that conventional CAD systems are now approaching to their final goal technically, though several future tasks are still remaining, for example, integration of the systems as a whole and improvement of man-machine communication.

But, all these facts do not mean designers can get creative or intellectual support from computers. We think there is an additional task, i.e., intellectualization of CAD systems for increasing productivity and for decreasing errors; the more complicated and the larger products become, designers will need more computer supports for the full scope of the design process including thinking process.

We conducted a survey on several design cases to know what to rationalize in the machine design processes, and how to do so. Figure 1 shows one of its results, i.e., what stage is most time-consuming when a design process includes several stages as shown in Table 1. From this figure we can conclude that the conceptual design stage leads the field and that it needs rationalizations most urgently, for the detail design stage has been rationalized by introducing conventional CAD systems.

Let us consider the development of a completely new product. First, the designer has to think about many alternatives from the given specifications written in a language. He may begin with

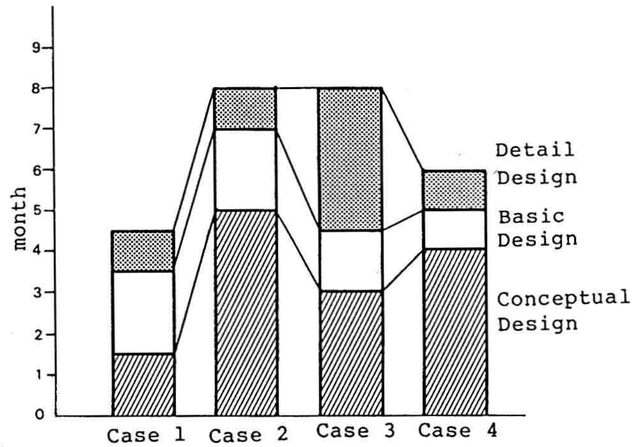


Fig. 1. Analysis of Design Process

Table 1. Stages of Design

STAGE	WORK
Conceptual design	Decide the basic method and basic structure of the design solution.
Basic design	Decide the layout and structure of the design solution.
Detail design	Decide the minute specifications of the parts.
Production design	Generate the necessary data for the production.
Prototyp-ing & test	Trial manufacturing and the test.

rough sketches or some notes to advance to more concrete ideas of the design solution. But, conventional CAD systems do not provide such facilities as drawing rough sketches supplemented by comments with essential meanings, nor can they check on the feasibilities of materializing of these ideas.

We think that the essence of designing indeed exists in the conceptual design stage, and therefore we stress in this paper how to utilize computers for that early design stage. Thus, as one of the functions of the future CAD systems, supporting designers in the conceptual design stage should be included.

2. DESIGNER'S WORKBENCH SYSTEM

First, let us begin with considering what the specifications for future CAD systems are. Generally speaking, we can point out three problems of conventional CAD systems as follows.

(1) Systems are not intelligent, e.g., sometimes they accept designers' inputs without checking errors or mistakes. Moreover, they cannot provide answers to designers' questions.

(2) Bad man-machine interface or communication problem. That is, first the designer has to feed into the computer the image of a candidate solution by commands or some other means; then he has to check if it matches his expectations or not. Because the amount of the data to be put in is normally huge, there are some possibilities of errors, mistakes, or misunderstandings during man-machine communication; and hence work efficiency is inevitably low.

(3) Non-integrated system environment. Normally, there are certain interface problems between two different CAD systems. These problems are mainly caused by bad interface design; but a much more important reason is non-integrated or non-unified data description. If a designer wants to use another system, he has to translate and again feed the data into another system manually. Though there are attempts for automatic data exchange [7], just a mechanical data exchange will cause a loss of the meaning. This is fatal for intelligent information processing that depends on the meaning.

From these problems we can derive some specifications for our future CAD systems.

(a) Integrated systems should support designers in all the design stages. There are two aspects of "supporting" designers.

   (a1) All the design works can be done on one system; for example, drawing, calculation, writing documents, data retrieval of patent information, simulations, planning for production, etc. This is called <u>integration of the systems.</u>

   (a2) All the information necessary for design can be obtained from the system, including patent information, information about designs done in the past, know-hows, and so on. This is called <u>integration of the knowledge.</u>

(b) The data description method should be integrated, and it should be unique and commonly used in all the designing stages. It is not enough to have adjusted interfaces corresponding to several data description methods. This is called <u>integration of the models.</u>

(c) The system should be intelligent.

   (c1) It should support designers <u>intelligently</u>, and the man-machine interface should be <u>intelligent</u>.

   (c2) It should provide <u>intelligent functions,</u> i.e., inference capability, so that even an unskilled designer can design fairly well.

This means that the system must understand the intents of the designer, detect errors or mistakes, suggest alternatives, answer to questions, etc. Among these functions, the most important one is error detection as early as possible to save time and cost.

(d) Naturally, the system should reduce time and cost for designing as a whole.

In order to materialize a future CAD system as mentioned above, we propose here the concept of a DWB (Designer's WorkBench) system shown in Figure 2. This concept is much wider than that of so-called CAE. This system features a very intelligent supervisor and a workstation for highly sophisticated man-machine communication. The designer exclusively works with this system, for it is connected to other computer systems by a LAN (Local Area Network) interface. For example, large-scale structure analysis programs may be executed on another big computer connected by LAN and not on the workstation.

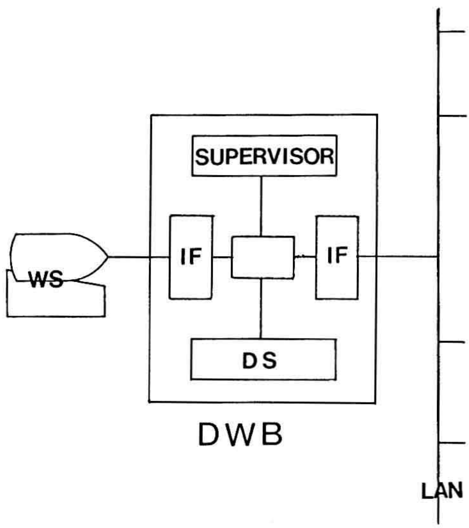


Fig. 2. Schematic Concept of DWB

The kernel of the system is the intelligent supervisor. This supervisor observes what the designer is doing as if it were watching through a window. It understands the designer's intents and actions and their results. Not only the supervisor but also other subsystems should be intelligent enough to solve design problems, so that, by using a DWB system, the designers can devote themselves only to decision making processes. What is important is that everything must be operated under an integrated or unified data description method.

Therefore, we must provide softwares as follows.

(1) Supervisor as the kernel for aiding designers.
(2) Intelligent man-machine communication or interface facilities.

(3) Design subsystems with an integrated data description method for aiding designers intelligently in all the design stages shown in Table 1.
(4) Documentation system.
(5) Drawing system.
(6) Systems for processing "rough sketches" or "notes."
(7) Consultation system for solving designer's questions.
(6) Data base facilities with an integrated data description method.

To present a method for realizing such a system with the above-mentioned softwares, we will continue the discussion in the following sequence. First, we assume a method that seems to be powerful and helpful for our purposes. Secondly, we will discuss how to introduce it into CAD systems for machine design, taking the issues of the design process into consideration. We will make use of some results obtained by general design theory for that purpose. Next, we will discuss "machine as a design object" from a viewpoint of representing it in a computer. Finally, we will reconsider the assumed method and check whether it is suitable or not.

3. INTRODUCING KNOWLEDGE ENGINEERING

Here we will discuss the realization method for DWB systems.

A DWB system should provide very intelligent inference and man-machine communication facilities to aid designers. At the same time, it shall have many design subsystems which can solve problems specific to machine design. It must also have an integrated or unified data description method. Therefore, the realizing method should be able to set up a large and flexible software development and implementation environment, so that system developers can write computer programs with inference ability. Knowledge engineering is expected to be one of such methods, that is to say, it is a new program development environment for implementing so-called knowledge as it is. For instance, it will be fairly difficult to implement knowledge which is written in mathematically exact logic on a computer using a conventional procedural computer language and based on the conventional computer architecture. In this context, logic programming will provide a flexible solution. Machine design includes knowledge that cannot be described in a procedural way but can be described only in a declarative or illustrative way. A good example is the mechanism choice problem in the conceptual design stage [1]. Moreover, machine design also requires knowledge that comes from skilled designers. The part layout problem at the conceptual design stage is a good example where heuristic methods of problem solving are effective.

Therefore, so-called knowledge based systems are indispensable for DWB systems. But before introducing them we must consider several general issues.

(1) Knowledge representation.
    (1a) Representation of machinery as design objects.
    (1b) Representation of knowledge about operations of the design objects.
(2) Operations on the design objects, i.e., inference and its control.

(3) Knowledge acquisition.

Fundamentally, first we must decide the representation before we discuss the inference or acquisition problem, but it will be difficult for the following three reasons.

(1) There may be many ways of representing machinery, and it may be changing corresponding to the progress of design.
(-- Diversity of the expression)
(2) Data quantities would be large.
(-- Bulkiness of the data)
(3) Most of design works lack uniformity, but still the expression must be multi-purpose.
(-- No uniformity of the expression)

From a practical viewpoint, we must pay attention to another important aspect. That is separation of the system implementor, the knowledge base author, and the user. Each of them plays an important role in a so-called knowledge based system, though they have so many different characteristics. The system implementor who makes the knowledge base system is not always an expert of machine design. The knowledge base author, who is an expert of machine design and writes the knowledge, is not always an expert of knowledge engineering. The user knows nothing about the system and perhaps knows a little about machine design. This aspect is important when we think about the system specifications.

Describing things in any given language requests us to make our viewpoints clear. That is, we need epistemology. This will be obtained from a design theory. Therefore, we will discuss the first problem, knowledge representation that needs epistemology, in Chapter 5, after we discussed the second problem, inference and its control, in the next chapter based on general design theory.

4. THEORY OF DESIGN AND DESIGN PROCESS

4.1. General Design Theory

Yoshikawa proposed general design theory as a guiding principle for establishing the foundation of CAD systems [2]. This theory is based on axiomatic set theory. We can derive theorems that explain design activities scientifically from the following three axioms. (See [2] for the definition of the terms). And it defines an integrated epistemology for the representation of the design knowledge as well.

AXIOM 1 (Axiom of recognition) Any entity can be recognized or described by the attributes.

AXIOM 2 (Axiom of correspondence) The entity set S' and the set of concept of entity (ideal) S have one-to-one correspondence.

AXIOM 3 (Axiom of operation) The set of abstract concept is a topology of the set of entity concept.

AXIOM 1 insists that the representation of an entity (including machinery, naturally) is given by its attributes. This means that an entity is described in an intensional or connotative way, otherwise any extensional (denotative) description would lose semantic relationship with the real world. On the other hand, in general design theory we define the concept of function as a subclass of the abstract concept that is derived from classifications of concepts of entity according to the meaning or the value of the entity. Here, classification might be done in an either subjective or objective way, but it may contain extensional or denotative expressions, because it is carried out by counting up of entities that belong to the same abstract concept. In other words, an entity is classified into some category according to its relative position to another entity, such that relationships of entities should be described. Hence, we need to get a representation method that allows both intensional and extensional descriptions.

AXIOM 2 insists that, if our knowledge about entities is incomplete (and this is the very case of the present state of our knowledge), the correspondence between entities and entity concepts has some failures (see [2]). When we build a knowledge based system, we have to check the completeness and consistency of the knowledge itself. This axiom tells that to do so we need somehow perfect knowledge. In a mathematical sense, it is enough to check the forms of the knowledge for completeness and consistency, as far as it is written in formal logic. However, once our system begins to deal with the actual world, it must also include "feasibility" checks besides completeness and consistency checks included in the mathematical world.

AXIOM 3 tells that we can operate concepts as in mathematics. But, we have to consider the "relationship" between the logical world and the real physical world. In normal formal logic (natural deduction), let P be any proposition, and

$$P \text{ or } \sim P$$

is always true by the law of the excluded middle. But it sometimes happens that we cannot decide between true and false, unless we have another information or proposition Q to decide it (this is called intuitionistic logic). For example, suppose a conversation,

A: Is Tom married?
B: (I know Tom, but) I don't know (whether he married or not).

In this context, proposition P,

"Tom is married",

was not confirmed nor denied; but before the speaker B answers to this question, he wanted to have some information about Tom. Clearly, there are many propositions like this question in a design thinking process. For example, suppose a design of a car conveyor system. We must know the weight of a car besides the load to decide the output power of the motor. But this is not known before we fix the motor.

To sum up, we must consider the following three points for the representation and inference problem. They are the theoretical principles of the integrated data description method.

(1) Mixing of intensional and extensional representations.
(2) Checking of "physical feasibility" of the design solutions.
(3) Introduction of intuitionistic logic to the logic system of design.

4.2. Theory of Design Process

The problems of the inference and its control have a direct relationship to the theory of design process, because the characteristics of designer's thinking process will be observed in their behaviors.

General design theory also tells that a design process is an evolution process of the metamodels [3] (Figure 3). Here, let S be the design specifications, s be the design solution, and the design itself be the mapping from S to s. Usually, S is described by the topology of function concept on the entity concept set, and s is described by the topology of attribute concept. If we introduce intermediate stages corresponding to the progress of the design, the entire design process is considered to be the accumulation of evolution or detailization at each intermediate stage.

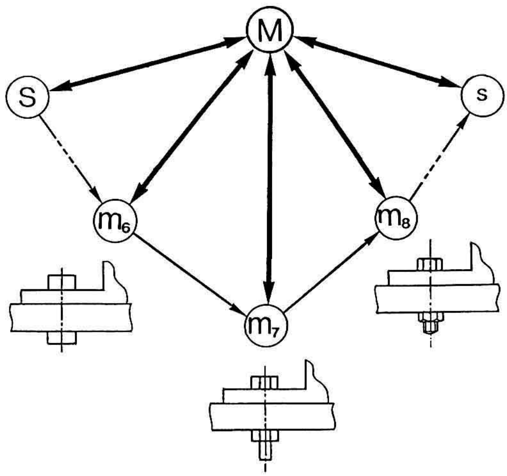
As the design process proceeds, the designer must confirm whether the candidate for the design solution satisfies the design specifications or not. This procedure is described as follows. Now, a metamodel M in a field f has functions F derived from its behaviors when it is materialized. Here, we define a model as

$$F(M, f).$$

The evaluation process e is the checking the value of

$$e(F(M, f)).$$

A metamodel is a model of a model or of a design object and is defined by finite attributes, which gives a framework for description of the design object as an entity. During this evolution process, the description of the metamodel will be detailed and the amount of its information will increase. The metamodel here produces actual model for evaluation and it will give the integrated or unified data description method.

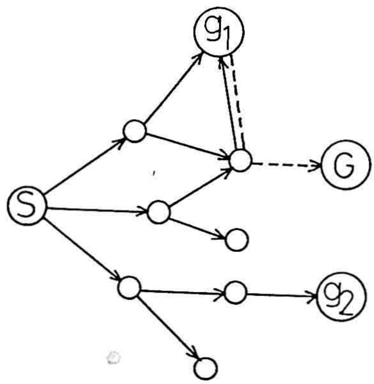


M: Metamodel
m: Design model
S: Design specification
s: Design solution

Fig. 3. Evolution Process of Metamodels

We can find out the correspondence between the above-mentioned theoretical design process and an actual design process. The conceptual design stage corresponds to setting up the structure of the metamodel, that is to say, to fixing a temporary design solution or goal. The basic design stage is, for example, an evolution process of the first design model to an assembly drawing through intermediate models, and meanwhile the metamodel is detailed. In the detail design stage, minute data of the product required for production are settled. In each process, designing progresses in such a way that first comes choosing a design model and then its evaluation. This means that the description of the metamodel and its contents change dynamically according to the progress of the design activity.

To sum up, first, the designer finds out a temporary goal that seems to satisfy the design specifications best. This process starts from the given data and goes forward. Secondly, she/he checks by evaluations if that goal satisfies the specifications or not. If it is not suitable, she/he has to change the old metamodel to a new one that seems to be much closer to the design specifications. If it is suitable, she/he has to figure out how to realize it. Both two processes include backward reasoning, from the given goal to its original causes. This is one of the characteristics of the design activities and is shown in Figure 4.



S: Start
g: Temporary goal
G: Real goal

Fig. 4. Thinking Process in Design

Another important fact is that, anyway, the design specifications are described in a language or an equivalent that contains information about the functions. On the other hand, the design solutions should be described not only in a language but also in geometrical data that should contain information about the attributes. This is a big gap. Moreover, knowledge that controls reasoning of design is clearly a mixture of experiential and logical knowledge. And, this knowledge used at each decision making is fragmentary, especially in the conceptual design stage, because choices are made by using very specific knowledge of the object and because it is almost impossible to establish integrated procedures that are valid for all the design objects. This means restructuring of the design knowledge is difficult in a uniform way. Especially, experiential or empirical knowledge is hardly written in a mathematically exact way, because it is too ambiguous sometimes.