Integrative knowledge-based design systems: A view

I. Petrovic

The IMS Building Research Institute, P.O. Box 834, 11000 Belgrade, Yugoslavia

Abstract

The paper describes a recent project which objective was the redesign of GIMSEXPERT, an existing architectural knowledge-based design system developed in 1987. Its critical generative problems appeared to be the rigid structure and limited evaluation criteria. The project's outcome is DESTOOLS, based on the "object-oriented-methodology", inspired by the traditional trial-and-error approach. It includes a set of interchangeable design methods that can be applied interactively by any desired sequence, producing or transforming a GIMS Building System object. Such "moderately-loose" system structure offers flexibility in use, avoids pitfalls of knowledge-based design systems with rigid structure, and is applicable in design research, education and practice.

1. PROTOTYPES DEVELOPMENT

1.1 Introduction

A "Design Research Workshop" was organized at the IMS Institute in 1987, in order to study the impact of computer technology on architectural design process methodology in general, and on possible qualitative improvement of architectural design in the Institute's IMS Building System in particular. The System consists of a prestressed concrete skeleton structure and various sub-systems, and has been applied extensively in housing and public building in Yugoslavia and many other countries.

One of the on-going research projects deals with elaboration of design tools to be used in the conceptual design phase of the IMS family houses. It was argued that in this design phase the critical decisions are made and expert advice needed most. The research content was also chosen for the practical reasons as the IMS design methodology had been sufficiently defined and tested, and amenable to computer modelling. The developed tools were expected to help and advice the novice IMS designers, and be used as teaching aids.

1.2 GIMS-EXPERT V.1.0

This is a consultant to a designer of individual family houses to be built in the "G(enerative)IMS building system, in this instance featuring the modular structural skeleton grid of 4,20 x 4,20 m. Following the user's preference of the shape of the house, type and number of functional spaces and the location data, the program generates a set of feasible design alternatives using the elements from the GIMS Catalogues and rules on their combinations, both contained in the Data/Knowledge Base. The catalogues include the "functional elements" which are the alternative functional entities, such as living rooms, entrance-cum-stairs-WC-or-bathroom units, bedrooms with corridor etc., all designed to fit the structural module. The ground-floor and first- floor plans are generated separately, and then fit together. The sketches are subsequently evaluated and sorted out according to the heuristics rules on the relation between indoor spaces and outdoor specific micro-climate conditions, vistas etc. Each design alternative can be explaned in view of the expert's evaluation method the design decisions are taken. The program module Roof Planner, which is a small production system by itself, produces all possible roof alternatives for all selected design alternatives.For each alternative, ground-floor and first-floor plans, and isometric or perspective surface-model views of the house with all possible roof solutions are produced. [1].

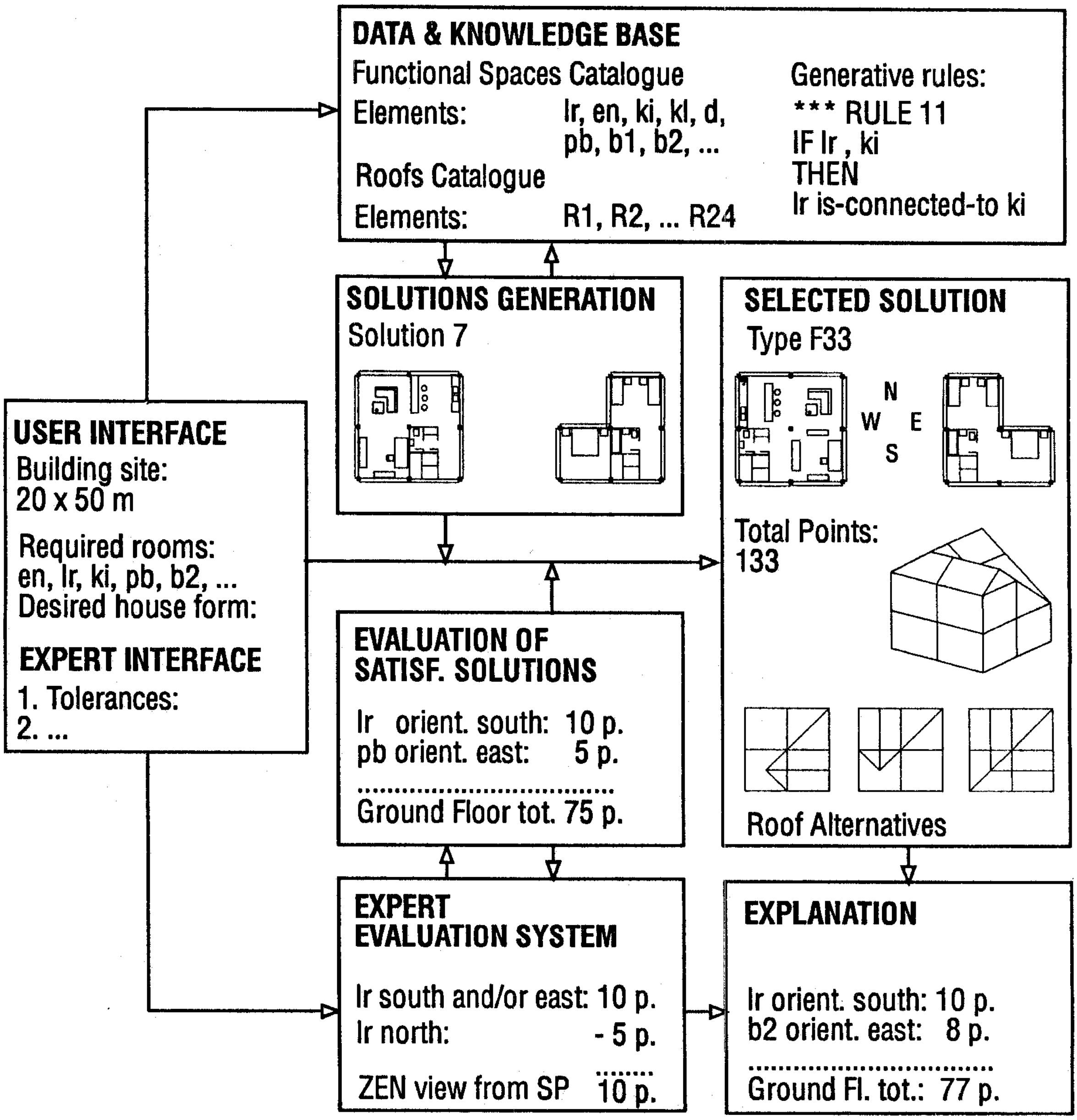

Figure 1. Diagram of GIMSEXPERT performance

The diagram explains the GIMSEXPERT performance on a very simple GIMS object which is meant to be only a "prototype" that can be elaborated to suit the particular circumstances. However, the program was tested in more complex real-life situation although the fully completed knowledge-base was missing. The project included two hundred dwellings. Combining a small number of dwelling types, a great variety of house designs was achieved by generation of different house shapes and roofs. Despite its seemingly "successful" initial application, in view of received comments by fellow architects and our own, it became obvious that the system needed an overhaul in order to answer the more complex requirements. In 1990 we decided to analyze the program, propose a redesign and eventually investigate the possibility of making an integrative design knowledge-based system.

2. THE PROBLEMS

Analyzing GIMSEXPERT, firstly we had to make a survey of the "generative" problems related to its performance, and decide on further steps to be taken. Many problem of various kinds and importance were thus encountered. Of these, we shall mention here only the problems related to GIMSEXPERT performance that were resulting from its structure, and to the employed means of creation and evaluation of the possible design solutions.

2.1 Problems related to design process structure

The existing GIMSEXPERT structure was modelled upon a well known design sequence consisted of problem definition), generation of possible solutions, and evaluation and selection of the proposed solution(s). Inasmuch the structure was envisaged to be iterative, in practice was not so; too many difficulties were encountered when "going backwards". Incorporation of new functional modules was possible in principle, but not a practical proposition. Obviously, the linear design sequence structure needed flexibility that was beyond its inherent nature.

2.2 Generation and Evaluation of Design Alternatives

GIMSEXPERT design solutions were restricted only to those that comply with some well-defined and structured conditions, and depended on the data and rules embedded in the knowledge-base. These, for example, included the default or newly defined relationships between various pre-defined functional spaces or any other specific user requirements. Evaluation and selection of solutions in such cases could be performed effectively.

However, if one wanted to select such houses that have a form which have a "good fit" with the built and natural surroundings, or to decide which house alternative is "nicer", the existing evaluation system was of no use; it had to include not only objective but also the subjective evaluation of the newly created design alternatives. Such new requirements asked for thr introduction of radically different approaches.

The user (designer's) participation in all parts of design process was seen not only as a means for introduction of new information and elimination of (almost inevitable) inconsistencies, but also as a "meaningful interface" between man and machine. All this indicated a need for the new system structure and method. We decided to follow our approach "from the practice", and reconsider again what was a design expert really doing when creating and assessing a design sketch.

3. DESTOOLS: A SET OF EXPERT DESIGNER'S METHODS AND TOOLS

3.1 A Concept of design process

What is the "proper" model of design process in the conceptual phase of design? What is the proper system structure?.

Let us assume that design process is the controlled transformation of the object-to-be-designed states. How is an object created? How is the transformation performed? A model of the traditional "trial and error" design method could be taken to be an "object-to-be-designed-oriented" process, whereby the object would be created rather informally, while the transformation of object states is achieved by application of various design methods and techniques from an expert designer's heuristic experience.

The similar concepts of an object as " the universe" and methods that are instrumental in its transformation are employed by O-OP, or "object-oriented programming" [2]. We could paraphrase and borrow some concepts of O-PP but not in their entirety. For a start, we can define an object by its data structure together with the methods capable of transforming the object state. The application of a method modifies the data stored in the particular real GIMS object. The design process is manifested through application of various methods on the object. An expert selects a method in relation to his objectives, intuition, design logic, and the nature of the task to be performed. In this sense, the problem of design could be equalled to the problem of selecting the proper design method, and the sequence of methods application. At present, we decided to leave the resolution of this problem to the designer's discretion. While O-OP focuses on the data to be manipulated rather than on the procedures that do the manipulating, we are equally concerned with both. Obviously, we are dealing here with an "expert's expert design system" where the control is performed by the expert. This does not exclude the partial control activities performed by the system.

The described concept of design process and its structure has the following essential features:

a) expert designer's selection of the method that suits the problem at hand and not vice-versa,

b) application of a method or their sequential mix as many times as necessary in any phase of design process,

c) object's transformation or replacement if it responds to the applied methods, and is approved by the designer,

d) inclusion of new methods, and

e) relatively simple methods of software maintenance.

The process is equally applicable to object classes (prototype solutions) and instances.

3.2 Building model

The object description includes the selected properties and features that characterize its behavior. In our case, we deal with GIMS houses. The object description has a specific hierarchical structure, and can be analyzed on the prototype (class) level or instance (specific application) level, which is at the same time a class level for the lower levels, and so on. The well known concept of inheritance in O-OP here is acceptable up to a point: it may not be applicable in case of semantic attributes of an object. Our interest has so far been focused on the conceptual class (building prototype) level. We are aware of the importance of the definition of the product model standard which is actually the meeting point of all our methods [3].

3.3 Methods and tools

What methods should DESTOOLS contain? Following our initial premises, we have decided to concentrate on such methods which a human design expert would consider as "critical". For example, they may be concerned with a problem of problem definition, creation of alternative solutions, their selection, etc., but can be very specific, according to the particular design approach of an expert. Which critical design decision is applied, is left to the expert to decide. In fact, on the conceptual level of design , many an expert applies only a few of the critical design activities. He may create a solution or two (most likely only one), and apply some of his favorite evaluations, being able to judge the appropriateness of each applied criterion and devise a plan for further prototype transformation upon such "partial" results. However, he might suddenly change his mind, and propose a new, completely different proposal, or select different evaluation tests. Completely defined and ordered knowledge-based design systems can hardly be expected to follow such suit.

At present, we are busy making a variety of tools that can be used by DESTOOLS. They perform various chores, such as generation of design alternatives from various points of departure, partial testing, etc. At this stage of development, we do not see the reason for their complete integration which would be a very complex task. Finally, we see such informal system organization as one of the ways to enable creative jumps in this early phase of design. If a designer can see a problem from many points of view, it is quite possible that his chances to sense an entirely novel design solution which, until then, might had been hidden "behind the obvious", are greatly increased. We expect that the future CAAD design systems shall have to pay respect to this and similar characteristics, considered "normal" in the traditional design process.

Other researchers have also defined the "classes of design and classes of design tools", claiming that "the different phases and types of design require a variety of conceptual and modelling aids", and proposed the following classes of tools: generic tools, parametric modelling tools, prototype editors, and shape grammar generators [4].

We propose a slightly different taxonomy to include: space synthesizers, evaluators and hybrid tools, categories that include the specific prototype tools developed or under development "from the beginning" in our Workshop. Obviously, the latter tools refer to a different level of generalization, and can be cross-referenced with the former. At a later stage of the project, we shall be able to do a more elaborate analysis of the nature and applicability of the developed methods and decide of the future strategic development.

3.4 Interface

DESTOOLS interface performs the following tasks: a) enables the user to activate a particular method or a sequence of methods, b) enables input and output of data, and c) transfers the data from one method to another as the need arises. Again, we find the difference between O-OP concepts and this simplified approach. In the text that follows, we shall mention only some selected tools that have been applied in the illustrating example, in PC DOS environment [5].

3.5 Space Synthesizers

ARCH. A constraints generator of all possible solutions of space configurations. Includes three types of space relationships: a) two adjoining spaces sharing a common wall where the relationship is based on the dimensions of the wall opening, b) two spaces without near proximity, and c) two spaces sharing a common wall but without the physical access between them.

3.6 Evaluators

OYSTER. A rule-production shell, allowing forward and backward chaining, and inclusion of new information in form of "What if" command. Tested on an number of "small" expert system applications. At present, it is used for the additional assessment of the selected attributes of objects, and for providing proofs of the object's attributes confirmation to the building regulations, or of any other hypothesis.

3.7 Hybrid tools

PDP-AAM. This is an unorthodox, neural net design research tool, based on the parallel distributed processes back-propagation algorithm [6]. Its properties include "learning" from examples, generalization of learned knowledge, and association of patterns. The semantic differential technique is used as an input/output device. The tool enables the following:

a) the subjective assessment of object's form,

b) learning from examples and producing a semantic differential for an unknown building form, and

c) learning from examples and producing a building form for the given semantic differential. A 3-d surface modeler is attached to the neural net, enabling the instant visualization of design proposals [7].

3.8 An illustration of DESTOOLS application

The mentioned prototype tools address only some design factors, but are efficient in their limited domain and quickly provide a set of building form alternatives, or evaluate solutions by the limited (but changeable) set of design criteria. They all have the standardized inputs and outputs amongst themselves and with the "Little 3-d Modeler", a program that produces isometries and perspectives.

The diagram that follows describes a test application of DESTOOLS concept on a GIMS object. As said, only a handful of selected methods are applied, yet they are capable of performing a variety of design procedures.

Let us start with the method ARCH and apply other methods clockwise. For the given number and types of GIMS functional modules and some default relationships, and the desired horizontal shapes of the house, the program produces alternatives of ground- and first-floor diagrammatic plans. Using OYSTER, we can apply a particular expert's evaluation in the specific location and select the plan with the highest score. Little Modeler produces 3-d representation of the chosen object. After that, we can make a subjective assessment of the GIMS object using PDP AAM program. Finally, OYSTER could conclude what the overall house character is, and explain the basis for the assessment.

The application of the methods can also be anti-clockwise. For example, we can start with the required house character, which will define the semantic differential. This shall in turn determine the 3-d volume and the plans. It is important that at any point the user can go back, set the new conditions, and experiment.

Finally we can start at any point of the design process, apply any existing method (or import a new one), apply it individually, or combine it with any other method in any direction. We may spend more time with a particular method and experiment with the semantic meanings of the house and search for the more acceptable ones, and then apply any other method of the set.

We see the present applicability of DESTOOLS is primarily in the field of research and education, although we hope to apply it in design practice in future. This shall depend on our decision of future use of hardware and software support.

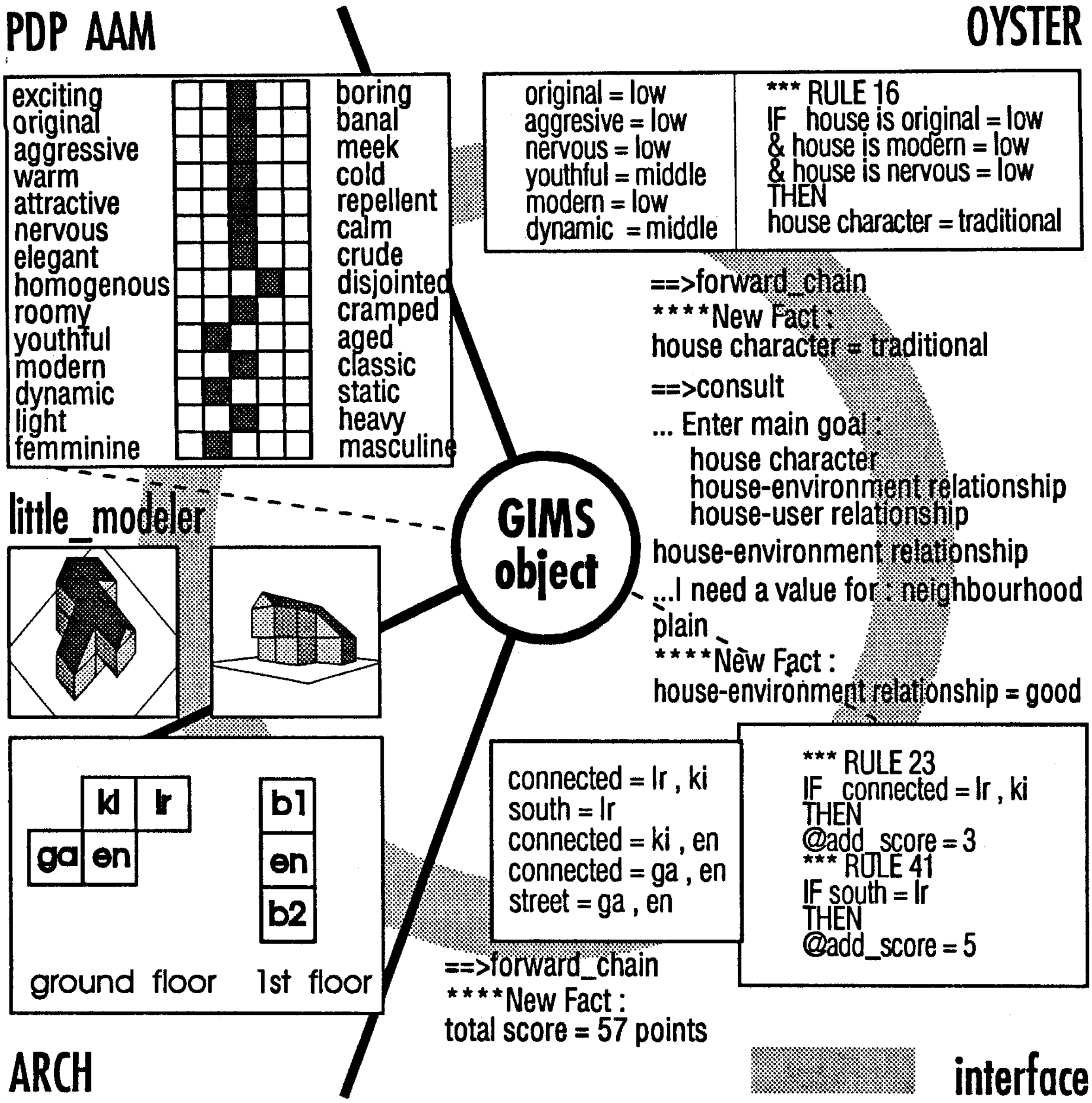

Figure 2. An example of DESTOOLS application

4. CONCLUSIONS

GIMSEXPERT is an automated designer, a "producer of alternatives", based on constraints generation of the possible solutions using a limited number of elements and rules from the data&knowledge-base. Inasmuch the program appears to "work", the number of assessing design criteria is small, the knowledge-base incomplete, and the program cannot deal with new information and the subjective assessment of the objects during the course of the process. The analysis of GIMSEXPERT indicated that the main problem generators were a) the design system sequential and noniterative structure, and b) the fact that a (inevitable) limited knowledge in the present-day knowledge-based systems must, by definition, produce the inadequate (not "limited") design if applied in uncertain context conditions typical in architectural design.

The first phase of the project produced a discrete design process, modelled upon a traditional design trial-and-error technique. It has an informal design decision structure centered on a design proposal, where the design process consists of "probing" the candidate solution with a limited number of independently applicable design tools adressing the "critical" design decisions.

The main difference between the integrated and nonintegrated design system is seen not in their performance but rather in the supporting design approaches and philosophies. Some process parts could be more formally defined and integrated if necessary, especially in the detailed design phase. The complete integration of a process of this type would be very difficult if justifiable at all.

The most important benefit from the work so far is in what we have learned about the nature of design process, and its relation to CAAD technology. Also, now we are aware of many more problems related to this type of work than we were at the outset of this project. This might prove to be an important factor in planning our future work.

5. ACKNOWLEDGEMENTS

Acknowledgements are due to Igor Svetel of thr IMS Institute who is the sole author of the following programs: ARCH, PDPAAM, and Little Modeler, and the co-author of the PC DOS application of OYSTER. Aleksandar Miric is the author of the shell OYSTER.

The project is financed by the research grants from the Scientific Council of the Republic of Serbia and the IMS Institute. The project director: I. Petrovic.

6. REFERENCES

1 I. Petrovic, M. Novkovic, M. Cubric, I. Svetel, and Z. Minjevic, GIMS-EXPERT: Consultant for Design of Family Houses in GIMS Building System, III International Conference on Research and Development of Building Technologies IMS'87, Belgrade, 1987, 227-245 (available from the author)

2 A. Goldberg and Robson, Smalltalk-80: The Language and its Implementation, Addison-Wesley, 1983.

3 B-C. Bjork, Issues in the development of a building product model standard, Building Research and Practice, No 1, 1990, 43-55

4 G. Schmitt, Classes of Design - Classes of Tools, in M. McCulloch, W.J. Mitchell and P. Purcell (eds.) The Electronic Design Studio, The MIT Press, Cambridge, Massachusetts, London, England, 1990. 77-90

5 I. Petrovic, Application of Research CAAD Programmes in Education, CAAD '90 International Forum, Hungarian Academy of Science, Technical University of Budapest, eCAADe, Budapest, 1990.

6 D.E. Rumelhart and J.L. McClelland (eds.), Parallel Distributed Processing, The MIT Press, Cambridge, MA, 1986.

7 I. Petrovic, Subjective Object Modeling in CAAD: Problems and Approaches, Second International Workshop on Computer Building Representation for Integration, Aix-les-Bains, France, 1991.