Object-Oriented Graphical Knowledge Base for Bridge Design

Nagoya University, Student Member, o Amin HAMMAD
Nagoya University, Member, Yoshito ITOH

1 Introduction

Designers use graphics intensively to express their ideas; therefore, graphics should be exploited as a deep knowledge representation means in CAD systems and not merely for drawing. In this paper a specialized drawing editor, using the object-oriented techniques, is suggested. The editor is used to build and update a 2-D graphical knowledge base of bridge components.

2 Requirements of Intelligent CAD Systems (ICAD)

In spite of the rapid progress of expert systems, they have been applied mostly for analysis and diagnosis problems. For synthesis problems like planning and design, the pure heuristic reasoning (shallow knowledge) that expert systems can offer is not sufficient and it is necessary to have access to the deep knowledge related to the problem.

During their professional carrier, designers accumulate a huge amount of detailed knowledge about the design space. In the phase of knowledge elicitation for building expert systems, some of these knowledge are considered as a kind of common sense for the experts and not expressed explicitly in the knowledge base.

Designers use drawings as a means to represent their ideas and to communicate these ideas among themselves and to other people. A simple draft can abstract a complex structure that needs many facts and rules to be described in full; therefore, graphical representation of design knowledge is an essential and natural component of any design expert system.

3 Role of Graphics in ICAD

In recent design expert system shells, graphics is used in the user interface to visualize the results in a manner similar to FEM post-processors.

CAD has comparatively an old history and has proved to be very useful for engineering design. A typical CAD program can be used for modelling the artifact to be designed, drawing it and doing the numerical analysis. The design data will be saved in a database for later use. A CAD drafting program is a collection of procedural functions that have a clear input and output and work in a pre-defined order. Some of these programs may offer some flexibility of tasks' reordering or have default values and allowable range for some variables, however the data structure they use can not accommodate the semantic relationships of the design object as can be done in knowledge bases. In other words, an expert system knowledge base has semantic relationships between design elements whereas a CAD system's database contains syntactic relationships among drawing elements.

Combining the expressiveness of CAD and expert systems is very promising for engineering design. Graphics can be used not only in the user interface, but as a central knowledge representation schema, side by side with the other schemata like frames and production rules.

4 Design Prototypes or Templates

Design can be classified into routine design, innovative design and creative design. In routine design, the designer can get advantage of previous designs similar to the one at hand by changing few parameters. Many researches mentioned the use of a kind of graphical prototypes for modelling common design patterns; for example Gross [1] proposed a knowledge based support for subsystem layout in architectural design.

Bridge design is a well established field where the majority of design activities fail under the routine design. Configuration of many types of bridges can be created beforehand as generic alternatives of design. The graphical representation of each component is represented as a generic meta-object with absolute and relative default dimensions and allowable ranges. Then components are combined to form sub-elements and sub-elements are combined to form elements in a hierarchical knowledge base. The constraints of each object and between different objects are imposed on the objects of the knowledge base to reflect the constraints used in design practice. The constraints considered here are mainly related to the shape of the bridge structure, some of them can be considered as common sense, others are imposed by design specifications or the availability of standardized components. Moreover, some constraints are purely aesthetic and related to designer preference. The shape constraints can be classified into three kinds:

- Topological constraints– define the different possible shapes, i.e. rectangle, arch, etc.
- Spatial constraints– like on-top, under, top-alignment etc. in addition to interference checking which prevents the overlapping of two components.

- Geometrical constraints– if we have two lengths, A and B, we can define the following default values and constraints:

  $Default(A/B)$, $Default(A)$, $Default(B)$,
  $Min_{A/B} \leq A/B \leq Max_{A/B}$, $Min_A \leq A \leq Max_A$, $Min_B \leq B \leq Max_B$.

These constraints can express, for example, the cross section of a bridge like the one shown in fig.1, and its components, like the I beam cross section shown in fig.2, parametricaly. In addition, they can describe the assembly rules and interface rules between different components .

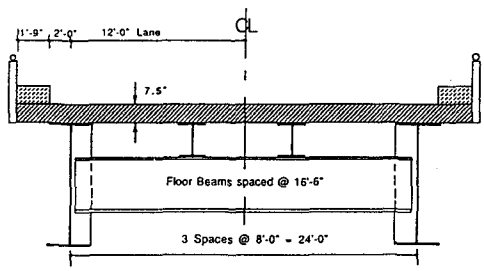

Figure 1: A bridge cross section

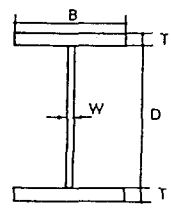

Figure 2: An I beam cross section

The resulting database can be used interactively in the preliminary design stage to help the designer in selecting the right configuration and adjusting the dimensions for a specific design without violating the constraints of the database. The resulting specific design will be saved in another database and can be interfaced to the subsequent structural analysis modules. All changes and feed-backs are reflected on the latter database that can be used for the final design documentation.

5 Object-Oriented Design of Structured Graphics

Structured graphics is useful for building applications that use a direct manipulation metaphor. Object-oriented languages offer inheritance, encapsulation, and runtime binding of operations to objects. Therefore object-oriented approach to structured graphics can give the benefits of both paradigms. Linton et al. [2], developed *InterViews* which is a graphical interface toolkit written using an object-oriented language (C++) and the X window library. *InterViews* includes a simple 2-D graphics editor that illustrates the application of object-oriented approach to structured graphics. This editor has many functions to create, move, rotate, scale and group objects in a MacDraw-like fashion. The specialized graphics editor suggested in this paper will be an extension of this editor by adding the ability to express the constraints discussed in the previous section.

6 Summary and Conclusion

Design drawings contains valuable information that should be used actively in ICAD systems. A large part of bridge design is a routine design. In this paper a specialized graphics editor was suggested to create a parametric graphical representation of bridge components in a hierarchical way and to built an incremental database that can be used to configure new designs. The authors believe that this software, when fully implemented, will help the designer in comparing between many correct configurations and therefore concentrating on the more conceptual design tasks. The software is written in C++ and runs on top of the X window library.

REFERENCES

1. Gross, M., Knowledge Based Support for Subsystem Layout in Architectural Design, in *Applications of Artificial Intelligence in Engineering V.* Gero, J.S. (Ed.), 1990.

2. Linton, M.A., Clader, P.R. and Vlissides, J.M., InterViews: A C++ graphical interface toolkit. *Technical Report CSL-TR-88-358, Stanford University*, July, 1988.