I－161 BriCode: SPECIFICATIONS' KNOWLEDGE-BASED BRIDGE DESIGN SYSTEM

NAGOYA Univ.	o	Student	Amin HAMMAD
SAITAMA Univ.		Member	Shizuo SHIMADA
NAGOYA Univ.		Member	Yoshito ITOH

1. INTRODUCTION

Design specifications change from one country to another and are revised when needed. In addition, parts of design specifications need interpretation by an experienced engineer. Provisions of specifications are usually 'hard coded' into the design systems which make it difficult to maintain these systems. To solve this problem, a knowledge-based object-oriented approach is used [1,2]. In this paper a bridge detailed design system using design specification rules is introduced. The system, called BriCode, is limited to the design of plate girder steel bridges with non-prismatic components according to AASHTO or Japan Road Association specifications for highway bridges. It should be possible to extend this work to the design of any other types of bridges or other codes easily just by increasing the knowledge base of the system. BriCode is developed using CommonLISP and KEE except for the mathematical programming module which is written in FORTARN.

2. KNOWLEDGE REPRESENTATION

An available model for representing a standard has evolved since the sixties [3]. The most recent model consists of four components:

Basic Data Items -- represent all the variables that have no explicit expressions in the standard defining their values.

Functions -- are algebraic or logical expressions and are used to compute some numeric or boolean values.

Decision Tables -- Decision tables provide a mean for the clear presentation of codes and a basis for automation of logical processes. A decision table is a concise tabular display of the logical conditions applicable in a given situation and of the appropriate actions to be taken as the result of the values of the conditions. Figure 1 shows a simple decision table named "check".

Information Networks -- There is a hierarchical relation among the data items in a design specification. Before some may be calculated, others (their "ingredients") have to have known values (the former are called "dependents" of the latter). The network representing the previous decision table and its dependents is shown in Fig. 2.

CHECK :

		R1	R2
Con.1	$\sigma_b < \sigma_a$	T	F
Act.1	O.K.	X	
Act.2	N.G.		X

T: TRUE
F: FALSE
X: FIRE ACTION

Fig. 1 - Decision Table Example

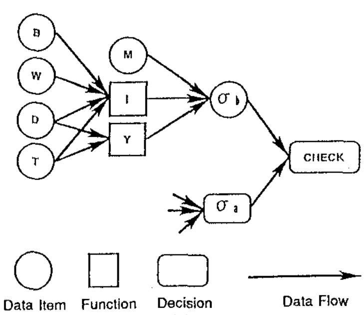

Fig. 2 - Information Network Example

In this study, an object-oriented description was followed. The elements envolved in the bridge design are considered as objects and the different attributes and processes needed in the analysis and design stages are implemented as attributes and methods for these objects.

The nodes of the network are represented as objects by assigning one unit for each node. The data related to that node are stored in the slots of the unit. The object for the moment of inertia of an I section and its attributes are shown in Fig. 3. The definition attribute is the LISP expression for calculating this data item using the dimensions of the section.

Object Name:	Moment of inertia of an I section
Type:	Function
Definition:	(+ (* B T) (/ (exp (+ D T) 2) 2) (/ (* W (exp D 3)) 12))
Ingredients:	(B T D W)
Dependents:	(SIGMA-B)
Value:	nil
Reference:	General-relationships
Explanation:	"Moment of inertia of I section, where B and T are the Breadth and thickness of the flange, W and D are the width and depth of the web"
Unit:	cm^4

Fig. 3 - Internal Representation of Specification Data Items

3. SPECIFICATION PROCESSING AND COMPONENT DESIGN

The design of any component aims to satisfy some requirements depending on the design standard. In the following paragraphs the method of retrieving these requirements, creating design constraints set from them and then satisfying this set is described.

3.1 Requirements Identification

Each component has an attribute called "Requirements", the value of which is a list of the requirements to be satisfied in the design process. For example, the Plate-Girder-I-Section object has the following requirements list: (Check-Shear-Stress Check-Normal-Stress Check-Thickness-of-Flange-Plate Check-Web-Width). These requirements are code-dependent, but this way of organization makes it easy to retrieve the related decision tables from the specified code. In addition to the code-dependent requirements, one designer-dependent requirement is added for the sizing of the I section, its name is 'Check-Sizing.'

3.2 Requirements Processing and Constraints Generation

Constraints are found by computing the value of the terminal node of the network, given specific values for some or all of the initial values. A requirements decision table is processed symbolically. For each requirement in the requirements set, the condition part of the rules which make the requirements satisfied will be accumulated in the constraints set list. The constraints set will be then satisfied by mathematical programming.

3.3 Constraints Satisfaction

The constraint set is to be satisfied by determining the optimal value of the design variables within the solution space defined by the constraint set. The value of the design variables should at the same time optimize an objective function suitable for the problem. In this study, the mathematical optimization approach was adopted. The objective function is contained in a special slot of the component of interest. The default objective function is taken to be the minimum area of the section.

4. SUMMARY

In this study, the design specifications are represented as knowledge-bases and processed symbolycally. The result of this processing, which is a set of design constraints, is given then to a mathematical optimization program that finds the values of the design variables. This systematic approach of using design specifications proved to be very flexible. The object-oriented programming helped to a large extent in implementing the different representational models.

REFERENCES

[1] GARRETT, J.H., "Object-Oriented Representation of Design Standards," IABSE Colloqium on Expert Systems in Civil Engineering, Bergamo, 1989.

[2] Rosenman, M. and Gero, J., "Design Codes as Expert Systems," Comput.-Aided Des. Vol. 17 No. 9, 1985.

[3] Fenves, S.J., "Tubular Decision Logic for Structural Design," ASCE, J. of the Struct. division, Vol. 92, No. ST6, 1966.