VI-78

SPECIFICATION RULES KNOWLEDGE BASE FOR BRIDGE DESIGN EXPERT SYSTEM

NAGOYA Univ. o Student Amin HAMMAD
NAGOYA Univ. Member Shizuo SHIMADA

1. INTRODUCTION

One of the significant difficulties in implementing bridge design Expert Systems (ESs) is in acquiring the appropriate knowledge. However, design codes already represent knowledge in a codified form, albeit an ill structured one, and are therefore ripe for the application of this technology. Provisions of standards are usually 'hard coded' into the design systems. Removing those provisions from the checking routines and placing them in a readily accessible external Knowledge Base (KB) can result in many advantages regarding the maintenance, readability, documentation, and updating of the system, and creating many libraries of standards.

In this paper the development of software for automated processing of design standards using the decision table approach is described.

2. KNOWLEDGE REPRESENTATION

Knowledge can be represented in different forms, mainly, Semantic Networks, Frames, Predicate Logic, Production Systems and Decision Tables (DTs). The various forms are not mutually exclusive. For example Frames can be used to contain the domain KB and DTs to carry out the inference.

DTs have been shown to provide a mean for the clear presentation of the sequential and parallel aspects of complex logical problems concerned with codes as well as providing a basis for automation of logical processes [1].

Systems of interrelated DTs may be used to represent large and complex logical problems. At the highest level, Switching Tables or Network System can be used to describe the overall logic and reference the appropriate lower level DTs.

A DT is composed of sets of conditions, actions and rules. Thus unlike other representation forms, a decision table allows for a set of rules concerned with the same objects to be processed simultaneously. Coupling this property with the ability of the PROLOG language to handle the parallel processing architectures of the fifth generation computers is a strong motivation to explore the suitability of DTs in the ES environment.

3. DECISION TABLES

A DT 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 [2].

An example of a DT is given in Fig. 1. This table is a reconstruction of table 1.4.1 of Ref. 3. A DT consists of four portions. The upper_left portion contains a set of conditions which consists of datum_value expressions. The upper_right portion indicates whether a condition is satisfied (Y: YES or TRUE), or not (N: NO or FALSE), or immaterial condition(space), i.e. implicitly TRUE or implicitly FALSE. The lower_left portion contains a set of actions to be performed. The lower_right portion indicates which action is to be taken if all the corresponding conditions in a given column are satisfied, the 'X' signifies that the action is to be executed, whereas a blank entry signifies that the action is not executed.

				1	2	3	4	5	6	7	8	9	10	11	12	13
STRUCTURAL TYPE OF BRIDGE	PLATE GIRDER BRIDGE	WITH RC SLAB	L <= 10	Y	Y											
			10<L <=40			Y	Y									
			40 < L					Y	Y							
		OTHER TYPES OF FLOOR								Y	Y					
	SUSPENSION BRIDGE											Y	Y			
	CABLE STAYED BRIDGE													Y	Y	
	OTHER TYPES OF BRIDGE															Y
TYPE OF GIRDER	SIMPLY SUPPORTED & CONTINUOUS GIRDER			Y		Y		Y		Y		Y		Y		Y
	CANTILEVER GIRDER				Y		Y		Y		Y		Y		Y	
ALLOWABLE DEFLECTION (H)	L / 2,000			X												
	L / 1,200				X											
	L*L / 20,000					X										
	L*L / 12,000						X									
	L / 500							X		X						
	L / 300								X		X					
	L / 350											X	X			
	L / 400													X	X	
	L / 600															X

Fig. 1 Example of Decision Table

Rule 1 of the DT reads as follows: If the structural type of bridge is plate girder bridge with reinforced concrete slab and the span (L) is less or equal 10 m, and the type of girder is simply supported girder or continuous girder, then the allowable deflection will be L/2000.

4. SOFTWARE ORGANIZATION AND IMPLEMENTATION ENVIRONMENT

Specification provisions represented in DTs can be implemented in the following manner:

1. A set of related design provisions is expressed in DT format.
2. The DTs are translated into program codes.
3. The program codes are then linked to a generic processor for design checking purposes.

Strategies of use of the previous technique using sequential algorithm in a FORTRAN implementation can be found in [4].

The present system is under development on KS-301 UNISYS work_station using the Explorer environment, mainly Common LISP language,PROLOG language and the KEE shell. It will use the same general scheme as above, the suggested steps are as follows:

1. Translating parts of Ref. 3 into DTs format for prototyping experimentation.
2. Editing the DTs by the KEE shell.
3. Using the LISP language, a PROLOG code generator will transfer the DT into PROLOG predicates.
4. Generic processor for automated design checking.
5. Interfacing with available CAD programs.

5. EXAMPLE

The DT of Fig. 1 is translated into PROLOG and the first two rules are shown.

```
/* Table 1.4.1  Allowable deflection */

/* rule 1 */

allowable_deflection ("L/2000")
   if structural_type ("plate_girder_RC_slab_L<=10")
   and type_of_girder (simply_supported_or_continuous).

/* rule 2 */

allowable_deflection ("L/1200")
   if structural_type ("plate_girder_RC_slab_L<=10")
   and type_of_girder (cantilever).
```

6. CONCLUSION

As opposite to directly coding design specification, this paper proposed the alternative approach of representing standards as DTs, which proved to be an adequate model for codes automatic implementation. Using the advantages of existing technology of AI in symbolic and logic processing and fitting the parallel characteristic of DTs with the parallel processing ability of PROLOG can result in a generic and flexible system for implementing codes' provisions and creating and maintaining many libraries of standards.

REFERENCES

[1] Rosenman, M.A., and Gero, J.S., "Design codes as expert systems.", *Comput. Aided Des.*, Vol.17, No.9, pp. 399-409, Nov., 1985.

[2] Rasdorf, W.J., and Wang, T.E., "Generic design standards processing in an expert system environment.", *J. Comput. Civ. Engrg.*, ASCE, Vol.2, No.1, pp. 68-87, Jan., 1988.

[3] Specification for Highway Bridges, part II (Steel Bridges). Japan Road Association (English edition). p. 1, Mar., 1987.

[4] Cronembold, J.R., and Law, K.H., "Automated processing of design standards.", *J. Comput. Civ. Engrg.*, ASCE, Vol.2, No.3, pp. 255-273, Jul., 1988.