Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 48 (1985)

Artikel: Standards representation and processing

Autor: Fenves, Steven J. / Garrett, James H. Jr.

DOI: https://doi.org/10.5169/seals-37449

Standards Representation and Processing

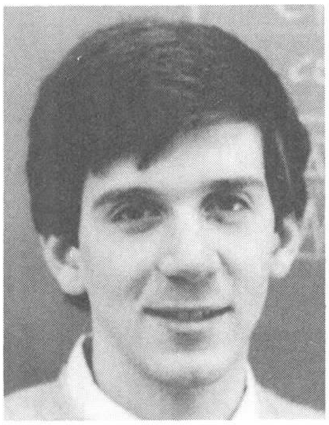
Représentation et traitement des normes

Richtlinien für Entwurfsnormen und deren Anwendungen

Steven J. FENVES
Professor
Carnegie-Mellon University
Pittsburgh, PA, USA

Dr. Steven Fenves received his Ph.D. from the Univ. Illinois, where he taught until 1972. His research activities center around computer-aided engineering, with an emphasis on expert systems, standards processing and databases. At Carnegie-Mellon Univ., he has been Head of the Dep. of Civil Engineering and Dir., Design Research Center.

James H. GARRETT, Jr.
Research Assistant
Carnegie-Mellon University
Pittsburgh, PA, USA



Jim Garrett received both his B.S.C.E. and M.S.C.E. from Carnegie-Mellon Univ. and is currently pursuing a Ph.D. in civil engineering at C-MU under the direction of Dr. Fenves. His Ph.D. thesis topic involves the development and implementation of a prototypical interface between design standards and CAD programs.

SUMMARY
This paper summarizes the formal representation of standards that has evolved over 15 years, starting with the 1969 AISC Specification for the Design of Steel for Buildings. The paper concentrates on recent developments in two types of standards processing, namely, for standards formulation and for standards use in a computer-aided design environment. An important aspect of the developments described is the incorporation into standards processing of the rapidly developing area of knowledge-based expert systems.

RÉSUMÉ
Cet article résume les travaux des 15 dernières années dans le domaine de la représentation formelle des normes. L'article met l'accent sur les développements récents pour deux types de traitement des normes, pour la formulation de celles-ci et pour leur utilisation dans un environnement CAO. L'utilisation des systèmes experts pour le traitement des normes constitue l'un des aspects importants des développements décrits dans cet article.

ZUSAMMENFASSUNG
Dieser Beitrag fasst die fünfzehnjährige Entwicklung formaler Richtlinien, beginnend mit der AISC Entwurfsnorm im Stahlbau aus dem Jahre 1969, zusammen. Der Vortrag bezieht sich besonders auf neue Entwicklungen, die zwei Gebiete der Bearbeitung und Anwendung von Richtlinien betreffen: Formulierung von Richtlinien und deren Gebrauch im computergestützten Entwurf. Ein besonders wichtiger Gesichtspunkt in der hier beschriebenen Entwicklung ist die Anwendung von Expertensystemen (Knowledge Based Expert Systems).

1. INTRODUCTION

Standards and design specifications are the primary means whereby analytical, experimental and empirical results on adequate structural behavior are transmitted to practitioners. They also become legal documents defining, or at least limiting, adequate design practices. To serve this dual purpose, standards must be complete, clear, unambiguous and usable with ease and confidence. The term "use" increasingly means incorporation of standards provisions into computer programs for design, detailing, and conformance checking.

Formal representation of standards has two purposes. First, the representation can be used by standards development organizations to insure that a proposed or modified standard possesses the requisite properties of completeness, clarity, lack of ambiguity, and ease of use before the standard is issued or promulgated. Second, the representation can serve as the starting point in the development of computer programs incorporating specification provisions, largely eliminating individual interpretations and reducing much of the expense of modifying application programs when standards are updated, or when a program must process differing standards of multiple jurisdictions.

2. THE REPRESENTATIONAL MODEL OF A STANDARD

The current model for representing a standard has evolved over many years. Many modifications were made during this evolution. Research prior to 1979 was presented in [3] and [4]. The next section describes the current state of the model which includes the contributions of Rasdorf and Fenves [15], Harris and Wright [11], and Howard and Fenves [13].

2.1. Current State of Model

The model for representing design standards consists of four components:

1. **data items** representing every variable that is found in the standard;
2. **decision tables** representing the logic used to determine the values of data items;
3. **information networks** representing the precedence relations among the data items; and
4. **organizational system** representing the organization (arrangement and scope) of the standard.

These four components are discussed in more detail in the next four subsections.

2.1.1. Data Items

Data items, or datums, represent all the variables occurring in the standard. The total set of data items, plus the relations between them, are intended to contain all the substantive information in the standard. Data items are one of four value types: numeric, a member of an enumerated set, "satisfied" or "violated", or boolean.

2.1.2. Decision Tables

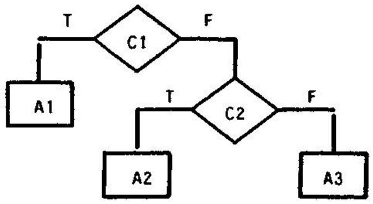
A decision table is an orderly presentation of the reasoning leading to the assignment of a value to a data item of the standard. Each table is responsible for producing a value for one, and only one, data item. A decision table is composed of conditions, actions, and rules, as shown in Figure 2-1a. A condition is a boolean expression that can only have the values of "TRUE" or "FALSE". As used in the model, an action can only be an assignment of a value to a datum. A rule is a prescription of a certain action, given that a specific combination of values of the conditions exists. A complete decision table for a datum gives an exhaustive set of rules for evaluating that datum.

Decision trees can be automatically generated from decision tables. They represent the same

	RULE 1	RULE 2	RULE 3
CONDITION 1 : F1(B,C)	T	F	F
CONDITION 2 : F2(C,D)	F	T	F
CONDITION 3 : F3(B,D)	F	F	T
ACTION 1 : A = E4(D)	X		
ACTION 2 : A = E5(C)		X	
ACTION 3 : A = E6(B)			X

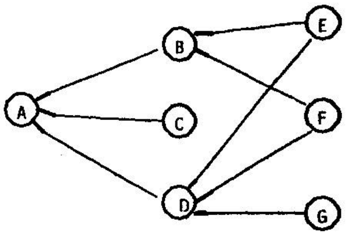
E1 - E6 are arbitrary expressions

a) DECISION TABLE FOR DATA ITEM "A"



b) DECISION TREE FOR DATA ITEM "A"

ASSUME: 1. E, F, and G are Basic Data Items
2. B depends on E and F
3. D depends on E, F, and G



c) ASSOCIATED INFORMATION NETWORK

Figure 2-1: Sample Decision Table, Decision Tree, and Information Network

information as found in the decision tables, but they also prescribe the most efficient order in which to evaluate conditions and choose an appropriate action. A sample decision tree generated from the decision table in Figure 2-1a is shown in Figure 2-1b.

2.1.3. Information Network

The information network is used to represent the precedence relationships between the data items of the standard. The network is composed of nodes, each node representing one data item in the standard, with each node connected to its ingredients and dependents by directional branches. The *ingredients* of a data item "A" are the data items needed to evaluate "A". The *dependents* of a data item "A" are the data items that have "A" as an ingredient.

An example of an information network is shown in Figure 2-1c. The left part of the information network for datum "A" shows that data items "B", "C", and "D" are needed to calculate the value of "A". However, the network does not show how the value for "A" is computed from its ingredients. It is the decision table that contains the precise relationship between the ingredients of a datum and the datum itself.

The information network can be used to:

- trace all the datums that have any possible influence on the datum in question , known as the **global ingredience** of that datum; and
- to trace all the datums that might be influenced by the datum in question, known as the **global dependence** of that datum.

Datums can also be classified according to their interrelations with other datums: *basic* datums have no ingredients, *derived* datums have both ingredients and dependents, and *terminal* datums have no dependents.

Rasdorf and Fenves [15] provided a method for expanding a datum within an information network into its subnetwork using the datum's associated decision table, as well as a method for compressing a subnetwork into a single higher-level datum. These methods provide a flexible interface between the information network and the decision tables.

2.1.4. Organizational System

Design standards are collections of provisions (terminal datums) which a design must satisfy. Each provision evaluates to either "satisfied" or "violated". The satisfaction of a provision depends on the satisfaction of one or more requirements, where a requirement has the form: <object> has <performance attribute>. The requirement's value is determined by a set of criteria contained within the standard.

The user of a standard must determine which provision(s) of a standard applies for the situation at hand. If this task is to be performed with the least amount of frustration and error, the standard must be organized and systematically defined such that this information is readily available. Decision tables only provide the logic within the provisions, and the information network only represents the precedence relations between the data items. An organization of the provisions is needed for accessing the individual provisions of the standard.

The standard provisions are placed into an organizational system in the following manner. First, the scope of the standard, i.e., the subjects covered, is decomposed into several trees of classifiers. A minimal classification system requires two trees of classifiers to represent the information contained in a standard: one for the physical entities covered and one for the performance attributes required for the entities [2]. In such a minimal system, all classifiers at the same level in the same tree must be:

1. mutually exclusive to guarantee that the selection of a provision according to associated classifiers is unique, and
2. collectively exhaustive to insure that every provision can be associated with its appropriate classifiers.

A more flexible classification system, developed by Harris and Wright [11], uses several independent classification areas, called *fields*. Each field is subdivided into a hierarchy of classifiers, called a *facet*. Partial facets from two fields are shown in Figure 2-2. The two fields are physical entities and limit states, respectively, where limit state refers to the failure mode of the physical entity. After the classifier facets have been built, appropriate classifiers from each facet are associated with each provision providing a medium for accessing the provisions.

Figure 2-2: Partial Facets of Classifiers taken from [11]

Howard and Fenves [13] developed a Generic Classification system for use in comparing standards. Their system uses facets of generic classifiers in order to classify all of the provisions in the standards being compared. They also developed a condensed information network which only shows the relationships between basic data items and requirements.

2.2. SASE - An Implementation of the Model

SASE (Standards Analysis, Synthesis, and Expression) is an integrated set of procedures being developed by the National Bureau of Standards (NBS) for computer aided formulation and expression of standards using the representational model presented [6]. The intended uses of SASE are to:

- *analyze* existing standards for completeness and uniqueness of its provisions, for connected-

ness and proper cross-referencing between provisions, and for completeness and clarity of the standards organizational system;
- *synthesize* new standards or modify existing standards by: providing a repository of information about the standard, allowing the standard's developer to analyze each provision as it is built, and allowing the standard's developer to analyze the partially completed standard; and
- aid in the *expression* of standards by providing several alternate frameworks for expressing the information content of the standard, such as outlines and indexes.

An expert system front-end is being planned for SASE to assist in the analysis of decision tables. The system will accept the conditions, actions and initially known condition entries, and then assist in completing the table based on dependencies among conditions. If this effort is successful, other expert systems may be attached, e.g., to assist in the ordering of information networks and outlines for clear and convenient expression.

3. RESEARCH IN STANDARDS PROCESSING

In this section, methods which use the representation presented above for processing standards in conformance checking and design are discussed. As will be shown, conformance checking is actually a subprocess of design.

3.1. Computer-Aided Checking of Standards

When performed manually by a novice designer, the process of checking design quantities for conformance with a standard expressed in conventional text form is a difficult task because:

- many of the applicable provisions are spread throughout the standard;
- proper cross-referencing among related provisions is usually not provided; and
- the performance attribute required is not explicitly stated.

The model presented provides a representation for orderly and efficient computer-aided standards checking. Several computer programs have been written that check designs for conformance with standards that are represented in this form.

In 1969, Goel and Fenves described a computer program that checks a design (i.e., a set of values for basic datums) for conformance with a standard represented by a network of decision tables [10, 1]. Goel's program requires that evaluation subroutines, i.e., subroutines used to evaluate the condition and action stubs (the expressions appearing on the left side of decision tables) of a decision table, be separately programmed in advance. A considerably modified program [18] contains a preprocessor that converts each stub into executable subprograms, leaving only the task of inputing the tables and the expressions appearing in the stubs to the user. A third program [16] extends the preprocessor by generating decision trees from the decision tables, producing source code IF-THEN-ELSE templates from the decision trees, and inserting the stubs into the source code, thus generating executable subprograms for evaluating decision tables. A fourth program, written in LISP [8], evaluates the conditions and actions of decision tables during execution, thus requiring no precoding or precompilation of the expressions. The treatment of the condition and action stubs is the only essential difference between the first, second and fourth programs. In these programs, a decision table is evaluated rule by rule. In the third program, the order of condition evaluation is prescribed when the decision tree is generated. In all four programs, when a data item value is needed during the evaluation of a decision table or tree and the data item is not yet bound to a value, the current decision table evaluation is suspended and the decision table or tree for the needed data item is evaluated. This evaluation method is called recursive, top-down, conditional, or backward chaining.

In 1984, Lopez began development of a knowledge-based system called SICAD which incorporates the representational model presented, a database management system, and an expert system application program which uses the checker and the database [14]. Besides the use of expert system and database technologies, SICAD differs from three of the previously described programs in that it expects that decision tables have been converted into decision trees and evaluates the decision trees.

3.2. Computer-Aided Design with Standards

This section discusses three research efforts in processing standards for the purpose of design. The first deals with the use of constraints in a data base, the second deals with methods for symbolically manipulating the provisions of a standard, and the third deals with a proposed interface between computer-aided design (CAD) programs and standards.

3.2.1. Standards Provisions as Database Constraints

The role of database management systems (DBMS) in structural engineering was presented in [5]. One of the key functions of DBMS is to enforce integrity constraints, that is, insure that the data present in the database are consistent with respect to these constraints. Furthermore, the relational DBMS model provides a methodology, called normalization, for removing dependencies among attributes of a relation. Examples of the various levels of normalization are given in [17].

A design database, containing information about an emerging design, must serve not as a passive repository of data, but as an active agent performing many of the consistency checks that are currently done manually. In some cases, it should also be able to do a limited amount of design, in the sense of assigning values to basic data items such that the applicable constraints are satisfied. One class of constraints deals with the satisfaction of the provisions of standards.

A mechanism of treating such constraints is presented in [7]. Briefly, the mechanism consists of the following components:

1. an additional constraint status attribute (with possible values of "satisfied" or "violated") is appended for each constraint on the attributes in the relation, and a function is provided to evaluate this attribute;
2. if desired, any constraint can be recast into an assignment procedure, which computes the value of a dependent attribute subject to the constraint and sets the constraint status attribute to "satisfied"; and
3. three commands are added to the DBMS: one to *invoke* the applicable constraint(s) on the current state of the database, flagging all tuples for which the status attribute(s) are "violated"; one to *activate* the constraint(s) for all future transactions, prohibiting any database modification that would violate the constraint; and one to *deactivate* constraint(s) if a design diverged or a new alternate design is to be started.

3.2.2. Holtz's Symbolic Manipulator of Standards

In references [1, 10], the passive or checking form of representing standard provisions is justified in the following manner. When performing design using a standard, different designers will choose different basic data items (dependent variables to be solved for subject to satisfying the applicable standard provision) to be the design quantity. Also, the same designer may switch the data item being designed at different stages of the design. Thus, the standard provision must be expressed in such a way that no preference is given to any one design method or sequence. For this reason, Holtz states, " A feature common to all constraints is that they are almost invariably formulated as a check on the adequacy of a design in terms of known or assumed values [12]." Holtz's work deals with the symbolic reformulation of such design constraints in order to produce bounds on the allowable value for certain basic data items, called "designable quantities" by Holtz. These bounds provide a range of values that gives a design which conforms to the standard, from which the user can choose. Symbolic reformulation of design constraints allows constraints to be used as "design" tools as well as "checking" tools.

Holtz developed a program, called CONMAN, that has three distinct uses, depending on the status of the basic data:

1. If all the basic data items have values, i.e. are bound, then CONMAN passively processes the standard and evaluates the network of decision tables in a manner similar to the programs described above.
2. If only one or a few data items are unbound and specified as designable, CONMAN asks which data item is designable and produces numeric bounds on the value of that data item such that any value chosen from within the bounded region conforms to the standard. The user then chooses a value from the feasible region and the program can proceed. This aspect of CONMAN is very useful for interactive design.
3. If all the data items are unbound and some are specified as designable, CONMAN produces symbolic bounds on the value of the designable data items. These symbolic bounds can be compiled into assignment procedures for repeated use. However, note that someone, i.e. the designer, must still specify how to choose a value from within the feasible region found by CONMAN.

The second use of CONMAN could automatically produce the assignment procedures for the database constraints discussed in the previous section. The third use of CONMAN could be used to generate automatically the standard-dependent portions of code in CAD programs. Holtz encountered difficulties and inefficiencies in the symbolic reformulation of constraints containing inequalities, making runtime reformulation impractical.

3.2.3. A Generic Design Standards Processor

A prototype standards processor is being developed that will act as an interface between CAD programs and design standards [9]. This proposed interface will take a component design or check request from a CAD program, perform the desired standard dependent task, and return the results to the CAD program. Knowledge of design techniques, materials, use of specifications, etc, will be represented using Artificial Intelligence and Knowledge-Based Expert Systems Techniques such as scripts and rules. This knowledge will be used by the standards processor to perform the CAD program's design requests.

The standards processor will call pre-made subprograms that perform the more common design and checking tasks. When a unique design task is requested, the standards processor will call upon an expert system to generate an executable task from existing subprograms and available knowledge. The expert system will also be capable of regenerating the executable subprograms for common design tasks when the version of the standard is modified. Thus, the standards processor can provide the processing efficiency of procedural programs, but can also provide the versatility of knowledge-based systems.

This separation of standard dependent knowledge (in the GDSP) and standard independent knowledge (in CAD program) has several advantages:

- changes in the standard produce no changes in the application CAD program;
- the standard need only be coded once for use by the generic standards processor, making it feasible to have an expert interpret and translate the standard provisions; and
- the application program can be made valid for any standard by simply changing the data, i.e., standard dependent knowledge, which the standards processor uses to satisfy the application program's requests.

References

[1] FENVES, S. J., E. H. GAYLORD, and S. K. GOEL, *Decision Table Formulation of the 1969 AISC Specification*, Civil Engineering Studies SRS 347, University of Illinois, August 1969.

[2] FENVES, S. J., K. RANKIN, and H. K. TEJUJA, *The Structure of Building Specifications*, Building Science Series 90, National Bureau of Standards, Washington, D. C., September 1976.

[3] FENVES, S. J. and R. N. WRIGHT, *The Representation and Use of Design Specifications*, Technical Note 940, National Bureau of Standards, Washington, D. C., 1977.

[4] FENVES, S. J., "Recent Developments in the Methodology for the Formulation and Organization of Design Specifications," *Engineering Structures*, Vol. 1, pp. 223-229, October 1979.

[5] FENVES, S. J. and W. J. RASDORF, "Role of Database Management Systems in Structural Engineering," *Informatics in Structural Engineering*, International Association of Bridge and Structural Engineers (IABSE), pp. 229-242, 1982.

[6] FENVES, S. J., R. N. WRIGHT, and F. STAHL, *Introduction to the Analysis Synthesis and Expression of Standards*, Draft , National Bureau of Standards, March 1983.

[7] FENVES, S. J. and W. J. RASDORF, "Treatment of Engineering Design Constraints in a Relational Database," *Engineering With Computers,* Vol. 1, No. 1, to appear in 1985.

[8] GARRETT, JR., J. H. and S. J. FENVES, *Decision Table Evaluator Users Manual*, unpublished internal course manual, Carnegie-Mellon University, 1984.

[9] GARRETT, JR., J. H., *A Generic Design Standard Processor for Use in Computer-Aided Structural Design*, unpublished Ph.D. Thesis Proposal, Carnegie-Mellon University, 1985.

[10] GOEL, S. K. and S. J. FENVES, *Computer-Aided Processing of Structural Design Specifications*, Civil Engineering Studies SRS 348, University of Illinois, November 1969.

[11] HARRIS, J. R. and R. N. WRIGHT, *Organization of Building Standards : Systematic Techniques for Scope and Arrangement*, Building Science Series NBS BSS 136, National Bureau of Standards, Washington D. C., March 1980.

[12] HOLTZ, N. M., *Symbolic Manipulation of Design Constraints - An Aid to Consistency Management*, Technical Report DRC-02-12-82, Design Research Center, Carnegie-Mellon University, April 1982.

[13] HOWARD, H. C. and S. J. FENVES, *Representation and Comparison of Design Specifications*, Department of Civil Engineering R-83-141, Carnegie-Mellon University, March 1983.

[14] LOPEZ, L. A. and S. L. ELAM, *SICAD: A Prototype Knowledge Based System for Conformance Checking and Design*, Technical Report, University of Illinois, 1984.

[15] RASDORF, W. J., *Design Specification Representation at the Outline-Information Network-Decision Table Interface,* unpublished Master's Thesis, Carnegie-Mellon University, August 1979.

[16] RUDNICKI, R., *Decision Table Based Software for Designing With Standards,* unpublished Master's Thesis, Carnegie-Mellon University, 1984.

[17] SCHAEFER, M. J., D. R. REHAK, and S. J. FENVES, "Introduction to Relational Databases Using Structural Engineering Examples," *Journal of Technical Topics in Civil Engineering,* Vol. 110, No. 1, pp. 1-18, May 1984.

[18] STIRK, J. A., *Two Software Aids For Design Specification Use,* unpublished Master's Thesis, Carnegie-Mellon University, October 1981.