1982

A methodology for the evaluation of designs for Standards conformance

Steven J.(Steven Joseph) Fenves
Carnegie Mellon University

A METHODOLOGY FOR THE EVALUATION OF DE3IGN3 FOR STANDARDS CONFORMANCE

by

Steven J. Farr/2S

December, 19J2

DRC-12-15-32

A METHODOLOGY FOR THE EVALUATION OF DESIGNS FOR STANDARDS CONFORMANCE[1]

Steven J. Fenves
University Professor
Department of Civil Engineering
Carnegie-Mellon University
Pittsburgh, PA USA

SUMMARY

A critical aspect of evaluation of designs is that of evaluating conformance with the governing standards and other regulatory documents defining acceptable designs. The paper presents a methodology for the formulation and use of such standards. The objective of the methodology is to assist developers in formulating clear, complete and unambiguous standards and to provide tools for generating CAD programs.

[1] *Presented* at International Association of Bridge and Structural Engineers symposium on 'Informatics in Structural Engineering.' Bergamo. Italy. October 1982.

1. INTRODUCTION

1.1 Role of Evaluation

To put the paper in proper perspective, a simplified model of the design process is first given. Design of a system, product or artifact in general involves three phases:

- *synthesis,* where one or more potential solutions are created satisfying a *few* key design constraints;

- *analysis,* where the performance of the candidate design(s) is computed and design parameters are selected so that the performance of the candidate design(s) is satisfactory - or even optimal - with respect to a *few additional* technological constraints; and

- *evaluation,* where the design judged to perform adequately is further evaluated with respect to *all applicable* constraints.

The design is considered acceptable if all constraints evaluate to *satisfied:* if any one constraint evaluates to *violated,* the design must be revised This simple model introduces two key issues. First, constraints are specified by groups or *classes.* The generic form of a class of constraints will be called a *requirement;* the application of that requirement to any particular instance is called a *constraint* [20]. Thus, the technological requirement in a flow network is:

$$\text{Flow} \ \hat{} \ \text{Capacity}$$

while the constraint on each component i is

$$\text{Flow}(i) \leq \text{Capacity}(i).$$

An integral part of the design process is to expand the given requirements into specific constraints for each instance of the class they pertain to.

Second, the design phase in which a given requirement is used is entirely up to the designer. The person (or agency) specifying a requirement does not know whether the designer will incorporate the constraints arising out of that requirement as a generative tool in synthesis, as a performance measurement tool in analysis or as a passive checking tool in evaluation. All requirements must therefore be given in a standard form. The most general form is the passive or checking form, that, is, a boolean expression evaluating to *true* or *false,* which can be interpreted as requirement *satisfied* or *violated,* respectively. This form can be used directly for evaluation, or it can be converted by the designer into active forms for use in synthesis or analysis.

1.2 Sources of Evaluation Requirements

Evaluation requirements, and the design constraints which they generate, come from three sources. *Technological* requirements arise from the physical principles governing the function of the artifact or system in question, such as conservation of energy, equilibrium of forces, compatibility of displacements, etc. These requirements are the easiest to represent and process, and in a CAD environment are generally incorporated into application programs or procedures. A second group of requirements are *internal* to the design process, and represent the owner's objectives and resources (e.g., the requirement "cost ≤ budget') or the designer's intention or style (e.g., "aspect ratio of a beam ≤ 2.0"). A third group of requirements is *external* to the designer or owner of a project, arising from the standards, codes, design specifications, regulations and other normative documents defining the acceptable performance or required characteristics of a system. Specifically, in an industry as widely dispersed and diversified as the building industry, building standards are viewed as the only "collective memory" of the profession [18]. Increasingly, regulations *are* introducing similar external constraints into many other design activities.

The remainder of this paper will deal specifically with the external evaluation requirements embodied in standards and codes. However, as the presentation will demonstrate, the methodology is equally applicable to internal requirements. For the purposes of this paper, the term *standard* encompasses all types of documents used for the evaluation of design and construction, including model and legal codes, consensus standards, and trade association and proprietary specifications.

1.3 Critique of Present Status.

The present mode of generating, promulgating and using standards suffers from two major deficiencies. First, there are no recognized formal methods for generating or reviewing the content or the form of proposed new standards or modifications of existing ones. Second, there are very few tools available for users of standards, that is, the designers responsible for producing designs conforming to the requirements of a standard and the regulatory agencies charged with enforcement of conformance. Both groups of users must exercise considerable effort in *interpreting* the written expression of a standard to generate their own evaluation procedures. The problem is further compounded in a CAD enviroment, where each organization, starting essentially from scratch, implements its own interpretation of a standard into a program for its own use. Even the slightest change in the standard requires changes, sometimes major ones, in all such programs. Neither the designers using these programs nor the persons who have to make judgments on the results generated have any direct way of ascertaining that the programs are based on the correct interpretation of the standard in questioa

1.4 Objectives of Methodology

The objective of the methodology to be presented is to improve design practice through better standards and better methods for the use of standards.

For the assistance of standard developers, the methodology applies to two distinct processes:

- *Formulation,* the generation of the information content of the standard; and

- *Expression,* the exposition of the information content in both conventional textual form and in forms adaptable to computer processing of the constraints in the standard

The methodolody provides some objective measures of two requisite properties of standards:

- *Completeness,* meaning that the standard can be applied to all possible situations within its scope; and

- *Clarity,* meaning that the interpretation of a standard can yield one and only one result when applied in any one situation.

For the use of standards, that is, the interpretation and application of standards in the evaluation of designs in both manual and computer-aided environments, the methodology provides a set of direct and convenient tools, as will be illustrated The presentation that follows is a brief summary of concepts developed over a ten-year period and applied to a number of standards, codes and specifications [2], [3], [14], [4], [5], [6], **[9].**

2. A MODEL OF STANDARDS

2.1 Provisions

The basic unit of a standard is a *provision* or normative statement Each provision has the function of assigning a value to a data item or *datum.* It is useful to recognize two kinds of provisions, distinguished by function:

- *Requirements,* or those provisions that are directly indicative of compliance with some portion of a standard Such provisions can normally be characterized by boolean data values, with *true* and *false* interpreted as *satisfied* or *violated.*

- *Determinations,* or all provisions that are not requirements. Such provisions are normally characterized by either numerical or logical values, including boolean, but are not amenable to characterization as satisfied or violated

2.2 Data Items

A data item or *datum* is a precise identification of an information element occurring in a standard The status (satisfied or violated) of each requirement is represented by a datum arid each result or variable generated by a determination is a datum. AH data assigned a value by a provision of the standard are termed *derived* data In addition, every other variable referred to in a standard but not explicitly assigned a result by some provision is a datum. Such data are referred to as *basic* or *input* data The list of data is similar to, but much longer than, a conventional list of definitions and symbols found in present standards.

2.3 Decision Tables

A *decision tabie* is used to represent the rules for assigning a value to a datum. A decision table is an orderly presentation of the reasoning leading to a decision. It is easily analyzed to assure that the reasoning leads to a unique result in each case and that no possibility exists for encountering an unanticipated situation.

The format and use of decision tables is best illustrated by an example. The following representative requirement is taken from Reference [1]:

"1.4.4 Site limitation for Seismic Design Performance Category D - No new building or existing building which is, because of change in use, assigned to Category D shall be sited where there is a potential for an active fault to cause rupture at the ground surface at the building".

Evaluation of this requirement will result in a value of *satisfied* or *violated* for the datum "Category D site limitation." The following data items are used in evaluating the datum:

- Seismic performance category (A, B, C, or D),

- Building stage (new or existing),

- Proposed work on existing building (true or false),
- Seismic performance category before proposed work (A, B, C, or D), and
- Potential exists for ground rupture from active fault (true or false).

Data that are used in the evaluation of a given datum are called the *ingredients* of that datum. Likewise, the datum is said to be a *dependent* of each of its ingredients. By itself, the list of ingredients for a datum does not give enough information to evaluate the datum; the decision table is used to collect all the rules for the evaluation of a datum. The decision table for the Category D site limitation datum is shown in Table 1.

TABLE 1 - Decision table for sample provision

	1	2	3	4	E
Conditions					
1. Seismic performance category = D	N	Y	Y	Y	
2. Building stage = new	...	Y	—	N	
3. Proposed work on existing building = change of use and seismic performance before proposed work # D	...	-	Y	N	
4. Potential exists for ground rupture from active fault = true	...	N	N	...	
Actions					
1. Category D site limitation requirement = satisfied	X	X	X	X	
2. Category D site limitation requirement = violated					X

The four parts of the decision table are separated by the broken lines The *condition stub* in the upper left defines all logical conditions that have a bearing on the outcome, for instance, "1. Seismic performance category = D." The lower left portion of the decision table is the *action stub,* defining all possible actions that can be taken. Here, Action 1 states that the Category D site limitation requirement is satisfied and Actlon 2 states that it is violated

The *condition entry* in the upper right-hand portion of the table is divided into a set of *rules.* Each vertical column contains one combination of conditions that defines a rule. For instance, Rule 1, read down the column, applies when Condition 1 is false (N) and the other three conditions are immaterial L). Rule 2 applies when Condition 1 is true (Y), Condition 2 is true, condition 3 is false (designated by the minus sign; it need not be checked, because it is predetermined to be false by the outcome for Condition 2) and Condition 4 is false. Rule 5, labelled E (for *else),* corresponds to all other combinations of conditions not explicitly included in the preceding rules, such as all conditions being true. The lower right-hand portion of the table, the *action entry,* shows by an X the action appropriate to each rule.

The *decision tree* generated from the decision table shown in Table 1 is shown in Figure 1. The decision tree provides exactly the same information for Rules 1 through 4 as the decision table, but it also shows two additional combinations of conditions. These additional combinations represent situations included in the else rule of the decision table. -

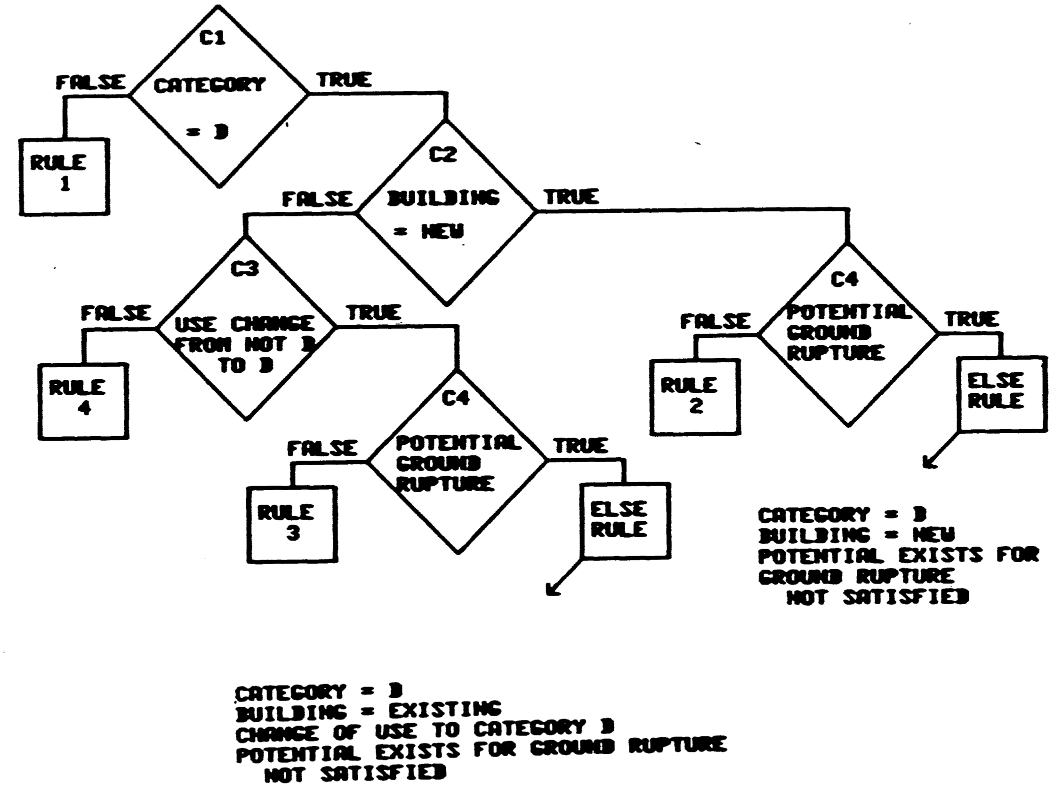

Figure 1. Decision Tree.
Each path represents one column of Table 1.

Occasionally a standard contains a single rule for the determination of a value. A decision table for such a datum would contain no conditions. Representation as a single statement, termed a *function*, is adequate.

2.4 Information Network

An *information network* is used to represent the precedence relations among the data in the standard. Each datum corresponds to a node in the network, and the nodes are connected branches that represent the ingredients of each datum. The information network graphically represents the flow of information through the data and thus the decision points in the set of provisions. Figure 2 shows such a network for a small portion of Reference 10. The figure shows that the determination of the required level of seismic analysis depends on the data items: "seismic performance category," "building configuration," "plan configuration," and "vertical configuration," which in turn depend on other data.

The entire information network can be assembled once each datum and its direct ingredients are known. The assembly is easily performed by a computer program.

2.5 Classification System

A *classification system* is used to generate outlines that represent the arrangement and scope of the standard. Requirements and determinations likely to be directly referred to by users are classified according to a model for provisions. The overall organization of a standard is based on a model structure for provisions and the classification of each provision according to that structure [11]. The model structure of a requirement includes two parts, a *subject* and a *predicate*. The subject may be a physical entity (for instance, a

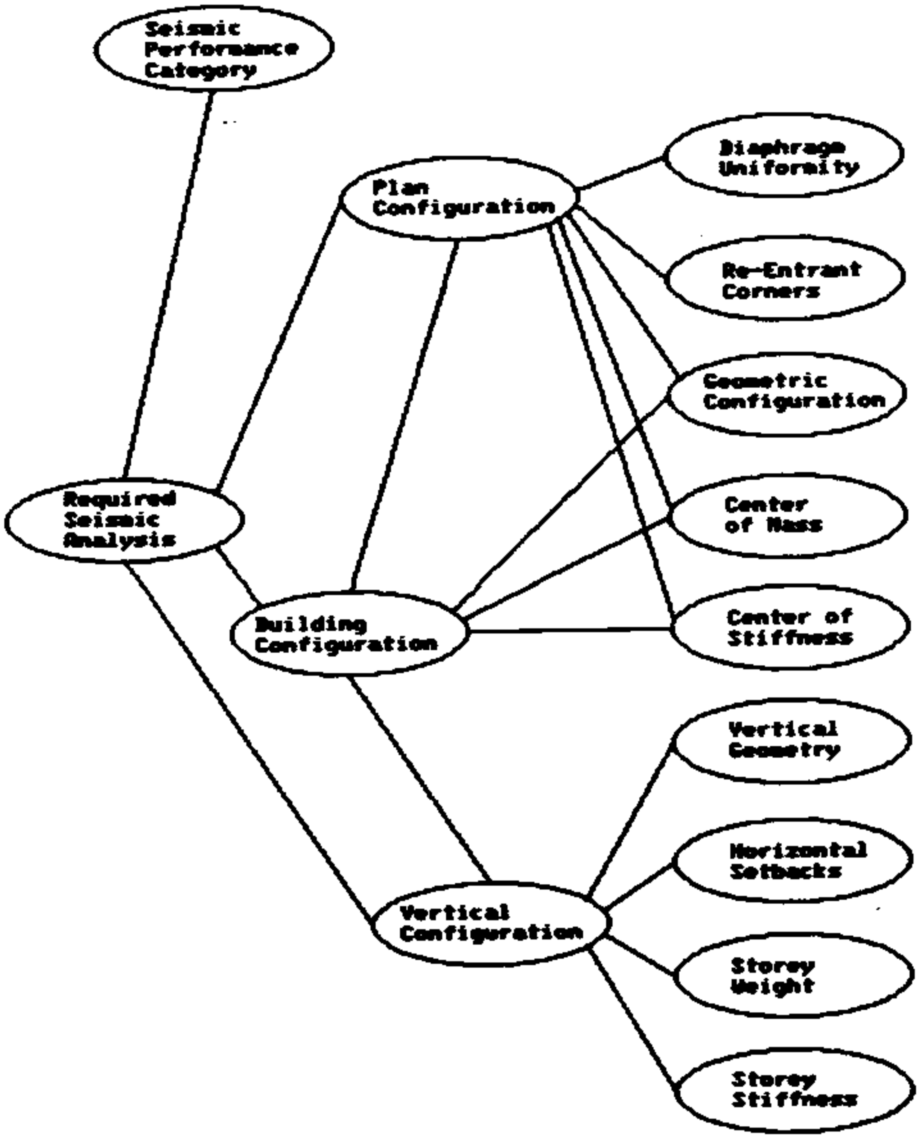

Figure 2. Information Network.
Branches are directed from the ingredient datum to the dependent datum.

part of a building), a process (for example, design or manufacture), or a participant in the process (for example, a designer, builder, or regulatory agency). The predicate is a particular quality required of a subject (for instance, strength or stiffness of a building part or quality assurance documents from a manufacturer). The list of classifiers pertaining to a particular provision is termed its *argument fist* (for example, "design" and "documentation" would be in the argument list for a requirement concerning the submission of engineering calculations).

The classifiers *are* systematically organized into hierarchies to represent the successively finer' subdivisions of the subjects and the required qualities (predicates) falling within the scope of a standard. Figure 3 provides an example of one hierarchy of classifiers; the example includes ail the subdivisions of the process of building design, which is one of the subject areas in Reference [10].

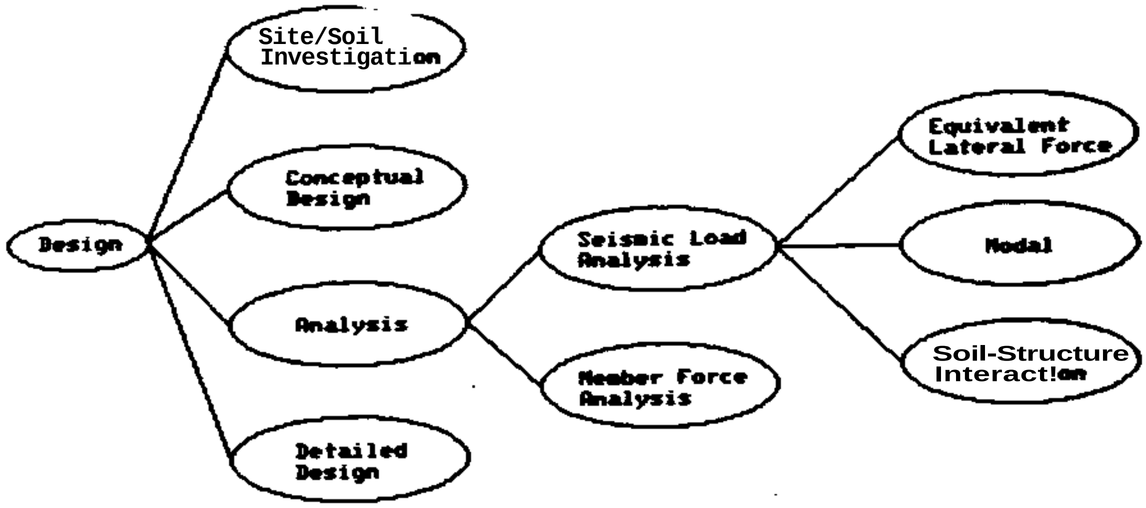

Figure 3. Classification Hierarchy for the Process of Design.

The provisions coming under a particular classifier are called the *scope list* of that classifier. The scope list can be generated by a computer program that transposes the argument lists for all the provisions.

3. APPLICATION OF METHODOLOGY

As indicated in the Introduction, the methodology is applicable both to the development of new or revised standards and to the use of existing standards. For the former, a distinction is made between formulation, that is, the generation of the information content and expression, that is, the presentation of that content These applications are briefly described in the following sections.

3.1 Applications in Formulation

Decision tables representing proposed provisions can be readily checked for *completeness* (all possible combinations of condition entries are included as rules), *lack of ambiguity* (no two rules can be matched simultaneously) and *redundancy* (two or more rules resulting in the same action). Of these, lack of completeness is most typical in early drafts of a provision

The else rule is a major tool in the analysis of provisions for completeness. Each combination of condition values included in the else rule must be reviewed to see whether a single action, such as Action 2 in the example shown earlier is appropriate, or whether the table is incomplete and needs additional rules to cover the scope of the provision completely.

The information network is useful in the analysis of the formulation of a standard because it clearly shows the impact of each datum on other data The information networks can be checked for *completeness* (absence of detached nodes or subnetworks) and the presence of *loops* ("circular definitions," where the evaluation of a datum requires the known value of one of its dependents).

In a similar fashion, the classification system can be checked for *completeness* (all provisions are classified in each of the relevant hierarchies) and for the property of *consistency,* that is, that uniform technical and logical bases are provided for comparable provisions.

3.2 Applications in Textual Expression

The purpose of expression is to present the information content of a standard \n a form convenient for use. For manual use, this means producing a textual form that is clear, consistent and easy to use.

To a limited extent, decision tables can be used to write the text of individual provisions, for example, by writing the text for simple or more common rules before that for the more complex or less frequent rules.

The information network is a major tool for organizing the text of a standard. The global ingredients can be used to order the written expression of a set of provisions. Each branch in the network corresponds to a link or reference that must be represented in the text Any branch not represented by close juxtaposition of the two data at either end of the branch automatically becomes a cross-reference between the two portions of the standard where the data are located Furthermore, two strategies of textual organization are possible. In the *top-down* strategy, the text is organized by giving the highest-level requirements first followed in turn by the lower-level requirements down to the determinations and eventually the basic data items; this gives the expert user the option to read only as far as he needs to, skipping those provisions which are familiar or known not to apply. In constrast a *bottom-up* strategy defines basic data first, then their dependent determinations, followed by higher-level determinations and eventually the requirements; this provides a "foolproof" step-by-step recipe which would be useful to the novice but would undoubtedly be repetitious and boring for the expert

Finally, the classification system provides the major tools for the synthesis of the organization of a standard Outlines can be developed by successively appending trees of classifiers from the hierarchies to produce a tree of headings resembling a table of contents. Different outlines can be obtained by varying the order in which the trees are appended Several trial outlines can be generated and the one best suited for the intended use of the standard retained Indexes are generated with classifiers as headings, usually in alphabetical order, and the scope list for each classifier provide a reference to the relevant provisions.

3.3 Applications for Computer-Aided Use

A number of existing or proposed standards and design specifications have been documented in the format described above. Formulations in this class include those for the AISC Specification for Steel Design [2], the ACI Concrete Code [13], the Tentative Criteria for LRFD Steel Design [15] and the Tentative Seismic Design Provisions [8]. Unfortunately, these formulations suffer from the fact that they have not been updated to reflect modifications introduced in the original yvritten standards.

The representation of standards in the form of networks of decision tables can be applied to CAD at four levels. At the first level of CAD application, the decision table formulations provide a convenient basis for programming segments of standards by conventional manual techniques, e.g., by coding the provisions in a procedural language such as FORTRAN. The primary advantages of using these formulations instead of the original written standard *are* first, that questions of individual interpretations are largely eliminated and second, that the required program logic - both for individual provisions and for their interrelations - is made much clearer.

At the next level (although to the author's knowledge this has not been done in a production environment) decision table preprocessors could be used directly. These preprocessors accept as input a combination of decision tables and procedural statements and produce as output source code resulting from an optimal conversion of the tables into sequences of IF-statements. [16] At the third level, efficient processors can be developed for checking conformance with standards provisions. Input consists of the data list decision tables, functions and the network represented by the ingredience lists of each derived datum. Just as in textual expression, two execution strategies are possible [3], [19]. In the *top-down* strategy, the program attempts to evaluate the topmost requirement specified by the user. If any of the ingredients are as yet undetermined, the program recursively descends and attempts to evaluate the missing ingredient If a basic data item is needed for the evaluation, it is requested from the user. Eventually, the program backtracks until it terminates by evaluating the topmost requirement This mode is primarily suitable for selective interactive "spot checking" of completed designs. By contrast in the *bottom-up* strategy, the basic data items are entered first and the derived data items are evaluated in sequence, without backtracking, until the topmost requirement is evaluated This mode is more suitable for routine evaluation of repetitive components in a batch mode.

The fourth level addresses the issue brought out in the Introduction, namely, that at the designer's option selected passive evaluation criteria need to be converted into active assignment procedures for use in synthesis or analysis. Thus, a simplified requirement on stress limitation in a structural element may be stated in a standard as

$$f = P/A \leq F$$

where f = actual stress
P = force on element
A = area of element
F = allowable stress.

A designer choosing an element area for a structural element for given P and F can do so subject to $A \geq P/F$. At other stages of design, the designer assigning a capacity to an element given A and F can do so subject to P^ FA. In other words, at different stages of design any of the data items appearing in a constraint expression may be designable subject to conformance with the requirement Methods of symbolic manipulation can be used to convert networks of requirements and determinations into expressions for bounds on designable data item as a function of the remaining data items [12]. The resulting expressions can be evaluated interactively, or they may be compiled into subprograms of CAD systems. It is worth emphasizing that the result is not automated design; the designer must still choose (or program the choice of) an actual value within the bounds allowed by the requirements of the standard-

It is to be reiterated that nothing in-the methodology presented or CAD tools described is specifically predicated on external evaluation requirements embodied in standards; internal requirements representing the designer's or owner's "standards" can be cast in the format presented and processed accordingly.

4. STATUS OF WORK

4,1 Aids for Formulation and Expression

The methodology for the analysis of standards was developed and refined over a numtfer of years by working with individuals and committees drafting various standards [6], [10]. The main shortcomings experienced were: first the analysts did not have sufficiently flexible computer-based tools to respond to the rapid pace of drafting and modifications; and second, there was a lack of long-term storage for the data (data item lists, decision tables, networks, classification hierarchies and outlines) between successive versions of a standard

As a result of this experience, the National Bureau of Standards (NBS) has commissioned a major software system, Standards Analysis, Synthesis and Expression (SASE) which provides a convenient user interface to enter, modify and display data, analysis capabilities (generation of decision trees, information networks, outlines and indexes), and a database for flexible storage and access [7]. NBS intends to provide training sessions and tutorial material for the use of the SPS system, and will make access to the system available to specification writing bodies.

4.2 **Aids For CAD Use**

Prototype programs have been developed for the top-down and bottom-up execution of networks of decision tables [19] and for the symbolic reformulation of passive checking requirements into expressions for the bounds on designable data items [12].

Both sets of programs accept a "high-level" description of the applicable standard, namely a network of decision tables. Thus, when the governing standard is updated or modified, only the resulting new decision tables are needed to re-generate the programs.

Both sets of programs are limited by the fact that, in the terminology of the introductory section they deal with requirements, not constraints. That is, they deal with generic data items such as "the force P," rather than specific instances, such as "the force P(i,j,k) on segment i of element j in loading condition k."

Work is in progress to develop general techniques whereby requirements can be "mapped" into constraints applied to instances of data residing in a database [17]. The major consideration is that such techniques be largely independent of the actual organization of the database. Modern database management tools, particularly the relational database model, can provide a large measure of this independence.

5. CONCLUSIONS

Standards, codes and design specifications embody hundreds, if not thousands of evaluation criteria which govern the acceptability of systems, artifacts and products, particularly in the building industry where codes have the force of law, intending to safeguard public health, safety and welfare. Furthermore, designers may choose key criteria for a priori generation, rather than a posteriori evaluation, of candidate designs. Standards and codes embody much of the "collective memory" of what has worked in the past; every major structural failure precipitates a search for code provisions which need to be added or modified to avoid similar failures in the future. Yet, designers overwhelmingly view standards as an imposition or impediment, frequently because of their awkward format and difficulty of interpretation, rather than their intent or content

In this paper, a formal representation of standards and a methodology for the for the use of that representation has been presented The methodology has two distinct applications:

- in the development of new or modified standards, it can assist in the formulation, by checking proposed standards for completeness and clarity, and in the expression of the content

- in the use of existing standards, it can assist in the generation of CAD programs incorporating evaluation and design procedures based on the standards.

ACKNOWLEDGEMENTS

Portions of this work were supported by the National Science Foundation and the National Bureau of Standards. The contributions of the author's colleagues R.M Wright and J.R. Harris and of numerous students is gratefully acknowledged

REFERENCES

[1] Applied Technology Council.
Tentative Provisions for the Development of Seismic Regulations for Bui/dings.
Technical Report, National Bureau of Standards, Washington, DC, June, 1978.
Special Publication 510.

[2] Fenves, S. J., Gaylord, E K, and Goel, S. K.
Decision Table Formulation of the 1969 American Institute of Steel Construction Specification.
Technical Report SRS 347, University of Illinois, Urbana, IL, August, 1969.
Civil Engineering Studies, SRS No. 347.

[3] Fenves, S.J.
Representation of the Computer-Aided Design Process by a Network of Decision Tables.
Computers and Structures 3(5): 1099-1107, September, 1973.

[4] Fenves, S.J., Rankin, K. and Tejuja, H.K.
The Structure of Bui/ding Specifications.
Technical Report NBS Building Science Series 90, National Bureau of Standards, Washington, D.C, 1976.

[5] Fenves, SJ. and Wright R.N.
The Representation and Use of Design Specifications.
In W.J. Hall (editor), *Structural and Geotechnical Mechanics,* pages 277-304.
Prentice-Hall, Englewood Cliffs, NJ, 1977.

[6] Fenves, S. J. f
Recent Developments in the Methodology for the Formulation and Organization of Design Specifications.
Engineering Structures 1:223-229, October, 1979.

[7] Fenves, S. J.
Software for Analysis of Standards.
In *Proceedings of the Second Conference on Computing in Civil Engineering,* pages 82-91. American Society of Civil Engineers, June, 1980.

[8] Harris, J. R., Fenves, S. J., and Wright, R N.
Analysis of Tentative Seismic Design Provisions for Buildings.
NBS Technical Note National Bureau of Standards, Washington, D. C.

[9] Harris, J.R, SJ. Fenves and R.N. Wright
New Tools for Standard Writers.
Standardization News 8(7): 10-17, 1980.
Philadelphia American Society for Testing and Materials.

[10] Harris, J.R, Fenves, S. J., Wright, R N.
Logical Analysis of Tentative Seismic Provisions.
Journal of the Structural Division, ASCE 107(ST8): 1629-1641, August, 1981.

[11] Harris, J.R.
Organization of Design Standards.
Unpublished Ph.D Thesis fUrbana: University of Illinois, 1980) , 1982.
To be republished as NBS Technical Note.

[12] Holtz, M. M, and Fenves, S. J.
Using Design Specifications for Design.
In *Proceedings of the Second Conference on Computing in Civii Engineering,* pages 92-101. American Society of Civil Engineers, June, 1980.

[13] Noland, J. L, and Feng, C C,
American Concrete Institute Building Code in Decision Logic Table Format
Journat of the Structural Division, ASCE 101(ST4):677-696, 1975.

[14] Nyman, DJ. and Fenves, SJ.
An Organization Model for Design Specifications.
Journal of the Structural Division, ASCE 101(ST4):697-716, April, 1975.

[15] Nyman, D.J., J.D. Mozer and SJ. Fenves.
Decision Table Formulation of the Load and Resistance Factor Design Criteria.
Report R-77-6, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1977.

[16] Pollock, S. L
Decision Tables: Theory and Practice.
Wiley-Interscience, New York, NY, 1971.

[17] Rasdorf, W. J.
Structure and Integrity of a Structural Engineering Design Database.
Technical Report DRC-02-14-82, Design Research Center, Carnegie-Mellon University, Pittsburgh, PA, April, 1982.

[18] Siess, C.P.
Research, Building Codes, and Engineering Practice.
AC! 56(111:1105-1122, 1960.
Detroit American Concrete Institute.

[19] Stirk, J.A.
Two Software Aids for Design Specifications Use.
Master's thesis, Carnegie-Mellon University, Pittsburgh, PA, 1981.
Unpublished M.S. Thesis.

[20] Wright, R. M, Boyer, L T, and Melin, J. W.
Constraint Processing in Design.
Journal of the Structural Division, ASCE 97(ST1):481-494, January, 1971.