COMPUTER REPRESENTATIONS OF DESIGN STANDARDS AND BUILDING CODES: U.S. PERSPECTIVE

S J Fenves, J H Garrett, H Kiliccote, K H Law, and K A Reed[2]

ABSTRACT: Standards representation and processing in the United States has had a long and interesting history of development. The work in the past has focussed primarily on representing a standard, evaluating the intrinsic properties of that represented standard, and evaluating designs for conformance to that standard. To date, for a variety of reasons, standards writing organizations and computer-aided design software vendors have not adopted much of the results of this research. The failure of the approach so far in the U.S. can be traced to two distinct areas. One major cluster of causes is methodological: the initial concepts were not backed up by usable, persistent computer tools; and the initial application and model were not representative. The second cluster of causes of failure is professional, and has a lot to do with the dynamics of interaction of individuals and organizations.

Future research must address the inadequacies of the current representations and create models that are able to represent all, or almost all, of the different types of provisions in any given standard; investigate and deliver a much richer set of processing functionalities, such as more support for use of design standards in earlier phases of design; support the treatment of multiple, heterogeneous standards available from distributed sources; and determine what type of support is needed to go from the textual versions of design standards to the formal models that can support sophisticated computation.

Keywords: Design standards, representation, processing, access, authoring.

INTRODUCTION

The introduction of computers into design practice has produced a marked dichotomy. Building design practice, and structural engineering practice in particular, is a prime example of this dichotomy. On the one hand, structural analysis programs quickly moved from specialized techniques, such as moment distribution, to increasingly general matrix and finite element methods. On the other hand, structural design and detailing programs have continued to be highly specialized, with each user organization or software vendor implementing its own interpretation of the governing design standard provisions into application programs. This "hard-coding" of design standards into design programs is a major barrier to the general acceptance and evolution of computer-aided engineering, as it does not provide designers having to make professional judgements the ability to view and understand the representations of the design standard on which the computations are based. Furthermore, any change in the design standard requires changes in all such programs. Therefore, researchers have been attempting to develop representations of design standards and procedures for computing with these representations.

While the initial research was motivated by the perceived need to develop representations on the basis of which computational tools may be dveloped and dynamically

[2]Steven J. Fenves, Sun Company University Professor, Department of Civil and Environmental Engineering, Carnegie Mellon University.
James H. Garrett, Jr., Associate Professor, Department of Civil and Environmental Engineering, Carnegie Mellon University.
Han Kiliccote, Research Assistant, Department of Civil and Environmental Engineering, Carnegie Mellon University.
Kincho H. Law, Associate Professor, Department of Civil Engineering, Stanford University,
Kent A. Reed, Leader, Computer Integrated Construction Group, Building and Fire Research Laboratory, National Institute of Standards and Technology.
The International Journal of Construction Information Technology, 1995, Vol 3, No 1, Page 13-34.
Note: Discussion open until 1 January 1995. The manuscript of the paper was submitted for review and possible publication on 28 February 1995.

maintained as the standards changed, in the course of the research two additional aspects of such computer representations emerged. One aspect deals with assisting designers in locating and accessing relevant standards provisions among potentially many applicable standards, regardless of whether these provisions are to be directly evaluated by computer tools or not. The second aspect is that the concepts and representations may be used by standard writing organizations themselves to develop the organization and textual representation of the standards.

This paper presents the history and current activities in the field, provides an assessment of progress to date, identifies some obstacles to further development still remaining, and outlines future plans for research. The presentation is based on a U.S. perspective. International efforts directly related to the concepts presented are briefly described. The authors make no claim for completeness of coverage; there may be other relevant U.S. efforts that have not come to the authors' attention.

HISTORY

Initial R&D

In 1966, Fenves made the observation that decision tables, a then-novel programming and program documentation technique, could be used to represent design standard provisions in a precise and unambiguous form (Fenves 1966). The concept was put to use when the 1969 *AISC Specification* (AISC 1969) was represented as a set of interrelated decision tables (Fenves et al. 1969). The stated purpose of the decision table formulation was to provide an explicit representation of the *AISC Specification*, which could then be reviewed and verified by the AISC specification committee and subsequently used as a basis for preparing computer programs.

Although the term "decision table formulation" became associated with the concept of a formal representation of design standards, it was evident that much more was involved than the tabular representation of the "if-then" rules of individual provisions. In particular, it became clear that the major sources of difficulty in interpretation, use, and programming were not in the individual provisions of the *AISC Specification*, each of which could be represented by a small number of decision tables, but in the lack of a consistent overall organization and the lack of cross-referencing between interrelated provisions. A follow-up project investigated possible bases for restructuring the *AISC Specification* (Nyman et al. 1973). The content of the specification was examined at four levels: the organizational network relating the requirements to be satisfied; the information network connecting the interrelated provisions; the detailed level representing the individual provisions in the form of decision tables; and the lowest level consisting of the basic data items referred to in the provisions. Alternatives for restructuring the specification were presented, but the study concluded that reorganization, without changing the content, was not warranted (Nyman and Fenves 1975).

As stated in the introduction, two separate aspects for computing with standards emerged around this time: computational support for authoring standards and computer-based design standards processing. Subsequent studies geared toward assisting standard writers in authoring design standards that essentially retained the four-level representation include (Fenves 1976; Fenves and Wright 1977; Harris and Wright 1980). This sequence of research activities culminated in a software system called SASE (Standards Analysis, Synthesis and Expression) developed by NBS (now NIST) (Fenves et al. 1987). SASE provided tools for creating and checking decision tables, information networks, classification systems, and organizations of provisions of standards. SASE required at least two facets in any classification system, one classifying the entities to which the provisions applied and the other classifying the attributes required by the provisions. Additional facets could be introduced as required by the subject standard. SASE also provided standards writers with the ability to develop and manage alternative or successive versions of a standard.

The original motivation of avoiding the need to recode design programs was put on a firm basis by Rehak, who first proposed a CAE (computer-aided engineering) system which would include a generic design standards processor operating on a formal representation of the governing design standard(s). In such a system, changes in a design standard would only require changes in the data (Rehak 1981). Subsequently, Lopez et al. implemented the SICAD system (Lopez and Elam 1984; Lopez and Wright 1985; Elam and Lopez 1988; Lopez et al. 1989), and Garrett developed the SPEX system (Garrett and Fenves 1987).

The SICAD (Standards Interface for Computer Aided Design) system was a software prototype developed to demonstrate the checking of designed components as described in application program databases for conformance with design standards in the SASE representation. In addition to a user interface, the prototype implemented three standards processors:

* The organizational network processor determines which provisions of a standard are applicable in the current context, based on the classification relevant to the context.

* The information network processor evaluates, in the current context, each of the provisions selected by the user from the list of applicable provisions, using the mapping processor described below.

* The mapping processor evaluates the data item requested by the information network processor using mapping functions into the application program database; if no mapping is found, the user is requested to give a value of the data item.

The SICAD concepts are in production use today in the AASHTO Bridge Design System (AASHTO 1994).

The SPEX (Standards Processing Expert) System used a standard-independent approach for sizing and proportioning structural member cross-sections. The system reasoned with the model of a design standard, represented using the four-level SASE representation, to generate a set of constraints on a set of basic data items that represent the attributes of a design to be determined. The only change to the SASE model was that the conditions in a provision were separated into applicability and performance conditions. The constraints were then given to a numeric optimization system to solve for an optimal set of basic data item values. SPEX implemented a standard-independent design process in that both the process of generating the set of constraints from the standard and the process for finding the optimal solution of these constraints were generic and thus not specific to the design standard being used.

Many other models for representing design standards have been proposed. Rasdorf and Wang developed a prototype knowledge-based system for design standards processing in structural design, in which the clauses of a design standard are represented as facts (Rasdorf and Wang 1988). The facts are treated as data on which generic rules operate, thus preserving the standard-independent nature of the processor. Topping and Kumar also proposed the use of facts to represent design standard clauses (Topping and Kumar 1989). Jain, et al., used predicates to formally represent the provisions of design standards (Jain et al. 1989). Rasdorf and Lakmazaheri discussed the use of first order predicate logic for representing and processing design standards for both conformance checking and design (Rasdorf and Lakmazaheri 1990).

Initial Aplications

As the AISC restructuring study presented above progressed, it became clear that the concepts and methods developed would find a major application in the authoring of new standards. The LRFD study by the Washington University research team, led by T. Galambos, provided an opportunity to test such an application. The application of design standards processing to what eventually became the AISC *Load and Resistance Factor Design (LRFD) Specification* (AISC 1986) took place in three distinct phases over a 10-year period. During 1972-76, Fenves provided assistance to the team developing the LRFD Criteria by reviewing successive drafts of the emerging Criteria for consistency and completeness and recommending organizations. The Criteria, dated October 1974, reflected most of the organizational suggestions made. Essentially no computer tools implementing the methodology were available: decision tables and organizational networks were generated and regenerated by hand. Such tools would have been most useful for maintaining the information over a long period of time. This experience contributed to the motivation for the SASE system.

In 1977, the AISC Specification Advisory Committee organized itself into ten Task Committees, charged with the conversion of the Criteria into the *LRFD Specification*. Nine task committees drafted specific portions, with the tenth serving as the Editorial and Coordinating Committee. The nine drafts differed considerably in organization and style, reflecting major changes from the Criteria. In 1979, Fenves was engaged by AISC to act as the editor for the Editorial and Coordinating Committee, charged with producing a consistently organized and worded draft of the entire Specification. Another cooperative period ensued, during which four successive drafts were generated over a two-year period, to be evaluated and modified by the Committee, often drastically. Again, no computer tools were available, requiring repeated manual reprocessing. In the third phase, AISC, with additional funding from National Science Foundation (NSF), sponsored the development of ELRFD, the electronic version of the *LRFD Specification*, described below.

A second initial application did not yield the same positive results. In 1975-78, the Applied Technology Council (ATC), funded by NSF and NBS, undertook the development of the *Tentative Provisions for the Development of Seismic Regulations for Buildings* (ATC 1978). This was a major cooperative effort, involving some 80 professionals representing the design professions, building code interests and the research community, intended to formulate a seismic design standard that would be applicable nationwide and be potentially incorporated in all major U.S. building codes. Fenves participated first in the Editorial Committee and later on the Coordinating Committee, applying the concepts previously developed to the overall organization.

During 1977-79, a follow-up study undertook an analysis of the ATC *Provisions* "to assist standards writers, designers, builders, regulators and all other users of the

Provisions by providing a formal representation of the logic of the provisions. The analysis was intended to serve as an aid in understanding, further development, incorporation in existing standards and codes, and use in design and construction practices" (Harris et al. 1979). The analysis documented 1200 discrete data items, 340 decision tables and an information network for each chapter as well as a global network. Quoting from the analysis report, "[R]ecommendations for effective cooperation are presented based on this initial experience. It is expected that the analysis results and techniques will be helpful to groups concerned with further development and use of the *Provisions*." Unfortunately, no further interaction occurred during the many succeeding iterations by ATC, the Federal Emergency Management Agency (FEMA) and the Building Seismic Safety Council (BSSC) in the development of several succeeding seismic safety standards.

In a joint demonstration project involving the University of Illinois, NIST, and the U.S. Army Corps of Engineers, the prototype SICAD system was used for checking reinforced-concrete structural components for conformance with provisions of the 1977 ACI Building Code Requirements for Reinforced Concrete (ACI 1977). The SASE model and Noland and Feng's previous decision table representation (Noland and Feng 1975) were modified to develop an executable SASE representation of chapters 8—12 of the code. A problem-oriented application environment was implemented that supported the description and conformance evaluation of structural RC-components. Mapping functions were developed for the input data items referenced in the ACI building code.

CURRENT MODELING ACTIVITIES

Formal Modeling Approaches

ELRFD

ELRFD was developed as an electronic version of the AISC LRFD Specification (Ackroyd 1988). In its present form, it is an interactive PC-based tool for conformance checking of structural elements. The user interactively completes two forms: one describing the structural component in terms of its geometric and material attributes and one describing the attributes of the applied loading. Facilities are available for storing and editing forms and for selecting attributes from stored tables.

The system consists of two major components. The organizational component uses the two forms to traverse a hierarchy of objects representing the scope of the specification to collect all applicable limit states (performance requirements) and detailed design requirements that have to be satisfied. The second component then evaluates the chains of linked decision tables corresponding to each requirement. At the conclusion of a conformance evaluation, the user has two options to view the evaluation results. In the browsing node, the decision tables can be traversed. If any requirement is violated, the conditions and actions leading to these requirements are shown in reverse video. The user can thus locate the input attributes that lead to the violation, edit the forms and resubmit the problem. In the report mode, an editable text report of the evaluation steps performed is generated.

ELRFD is the most completely integrated implementation of the initial research concepts described. The level of representation needed for a working program is probably an order of magnitude more detailed than the representation of the 1969 *AISC*

Object-Logic Model

An alternate model of design standards is the *object-logic model*, which combines two paradigms to represent and process design standards, using a unified object-oriented logic programming environment (Yabuki and Law 1993). This approach also allows the development of formal procedures to be formulated to check completeness, uniqueness and conflicts among the provisions (Yabuki 1992).

The system framework for the object-logic model consists of five basic modules (Yabuki 1992; Yabuki and Law 1993):

* A *standards base* consisting of a class hierarchy representing the organization of a standard and a set of method objects (a set of object-logic sentences) representing the individual provisions;
* An *object data model* to facilitate the linkage with a CAD model and to retrieve member attributes from an engineering database;
* A *conformance checking module* to check whether a given member is in compliance with applicable provisions and requirements. The reasoning mechanism is based on logic resolution and message passing among method objects;
* A *component design module* to select trial design sections based on a set of design heuristics and a generate-and-test strategy; and
* A *standard analysis module* to check potential problems related to completeness, redundancy and contradiction in the rule set represented by the method objects and to perform simple analyses on the provisions by checking whether the relationships among the method objects are connected and acyclic.

To illustrate the utilization of the model, a prototype system was implemented for the AISC *LRFD Specification*, using Prolog^{++}, MacDBI and Oracle software on a Macintosh computer.

Description Logic Model

While the object-oriented and object-logic approaches begin to address the need for a unified definition of the objects to which a standard applies, they only provide half of the needed definition: to properly use these models, a user must use the same classes of objects to represent the design as those used to model the standard.

Class-centered approaches, such as the object-oriented and object-logic models, rely on pre-defined schemas of classes which can be used to instantiate objects and modify them according to the rules enforced by the parent class. Automatic classification of an object in not possible because class membership conditions are not included in the definition of a class in class-centered approaches. By contrast, Hakim and Garrett proposed an *object-centered* approach that relies on the ability to create objects and then classify them based on an intentionally defined class schema (Hakim and Garrett 1993). In this approach, each class possesses an intentional definition in the form of the necessary conditions of member instances.

Hakim and Garrett used such an object-centred approach, implemented in description logic, for representing and using design standards. The description logic-based model uses intentionally defined objects (i.e., objects for which membership conditions are defined) to which requirements and data item definitions are associated. This approach allows a design standards processor to identify the object classes in the model of the standard to which an unclassified object belongs.

Context-Oriented Model

In the object-oriented, object-logic, and description-logic models, a class lattice is used as a method to organize the design standard; provisions are associated with the classes in this lattice. Because these associations between classes and provisions replace the conditional parts of the provisions, the representations of the provisions are significantly simplified. However, these models have a major deficiency: they require a very large number of classes and complex multi-parent subclasses to be created in the lower parts of the class lattice so as to represent all of the specific contexts to which the provisions in the standard apply.

The *context-oriented model* addresses this deficiency by extending the object-oriented, object-logic and description logic models. As with the previous models, the context-oriented model also uses the concept of class to organize the design standard. The provisions are directly attached to classes. Each class may have a number of subclassification hierarchies by which it can be specialized. Provisions attached to a specific class are tagged with the subclassifications from these classification hierarchies for which they are applicable (such a collection of subclasses is called a *context*, a term first used by Elam and Lopez to mean a collection of SASE classifiers (Lopez et al. 1991)). Many of the subclassifications in the classification hierarchies are provided with definitions that can be used to compute whether an instance of a class can be subclassified. By associating contexts with provisions, the need for large numbers of highly specialized subclasses from which to hang provisions is eliminated.

STEP and Product Models

Each of the modeling approaches described above assumed the existence of a product model for the artifact being designed. An essential difficulty in the integration of design standards with application environments is the lack of common models. In the SICAD prototype, for example, the mapping functions had to be custom-coded to match the application program's database structures; a new application program generally would require a new set of mapping functions.

There is an international standardization effort underway that may alleviate this difficulty. Known informally as the Standard for the Exchange of Product Model Data (STEP), and formally as ISO 10303—Product Data Representation and Exchange (ISO 1994), this standard was undertaken in the manufacturing industry in the mid-1980's to replace the various existing standards that dealt primarily or solely with CAD-system data structures rather than product models.

The fundamental premise of STEP is that product data schemata (e.g., the structure of product models) can be standardized and that only product data instances need be exchanged, accessed, or archived. This premise appears to hold in restricted domains such as manufactured piece parts but has not yet been seriously tested in extended domains such as the built environment.

In the STEP methodology, application domains define their information requirements in terms of application reference models (ARMs). The ARMs are interpreted in terms of standardized, integrated resource models (created separately), yielding application interpreted models (AIMs). The AIMs are then implemented using standardized methods for file exchange, data base access, and the like. All normative models, in particular, all AIMs, are expressed in a formal language known as EXPRESS. The parts of STEP that are implemented are known as Application Protocols, which are normative documents. A STEP Application Protocol consists of an ARM, an AIM, one or more implementation methods, an abstract test suite, and a usage guide.

The Architecture, Engineering, and Construction Team in the STEP activity is working in four areas: building construction, offshore, process plants, and shipbuilding. The shipbuilding subgroup is well advanced toward a series of application protocols that cover ship structure and various ship distribution systems such as piping, HVAC, and electrical raceways. The process plant subgroup has embarked on the development of several application protocols that cover the functional characteristics, schematic representations, and spatial configurations of process plants. In both subgroups, the application protocols primarily deal with detailed design and fabrication data. U.S. industry has made major contributions to these efforts. The building construction subgroup is just embarking on the development of application protocols dealing with structural steelwork and HVAC systems. It is also attempting to develop a collaboration strategy with other international standardization activities dealing with the classification of building products and processes.

Every major CAD vendor in the U.S. and elsewhere is participating in the development of STEP. If these vendors provide STEP-conforming interfaces to their application programs, then the task of integrating generic standards processors to their systems will be substantially easier; indeed, it could be partially automated. The caveat is that application protocols with scopes overlapping the scopes of the standards to be processed must be developed and be supported by the CAD vendors.

Text-Based Approaches

In addition to determining the compliance of a product design with the applicable provisions of a formally modeled design standard, users of design standards need considerable assistance in identifying applicable standards, and provisions within those standards, for a given design context. Researchers have thus investigated text-based approaches for helping designer find applicable provisions of design standards. The next two sections describe two such approaches.

Hypertext-Based Approaches

The use of hypertext for storing design standards documents in electronic form has been gaining popularity (Cornick 1991; Malasri et al. 1991; CIB 1992). Each section or provision of the standard is represented by a node. A table of contents is also a node, in which the user can access any provision by clicking the the provision title. Hypertext allows a user to access desired information without having to view unnecessary information. The electronic document can easily be updated while previous versions are retained and attached as reference. The Standard Generalized Markup Language (SGML) (ISO 1986; Goldfarb 1990) has been used to develop hypertext applications that are independent of any specific browser technology, but the developers and the users must agree on the use of certain SGML elements used for hyperlinks. HyTime (ISO 1992) is a standardized SGML application that explicitly defines an architecture for creating hypertext and hypermedia applications and deals with the problem of hyperlinking as a problem of addressing objects in space or time.

Reed outlined a framework for integrating different views of a standard into a unified model using SGML as the notation language (Reed 1991). Reed proposed combining the executable SASE model with a text model and building standardized hyperlinks to referenced standards and to application programs. The approach would allow standards writing bodies to issue one file that could be parsed variously to print a standard, to browse it and its referenced standards via hyperlinks, and to use the standard in SICAD-like implementations. The framework has not been implemented, and, in any case, needs to be recast in the form of HyTime.

The *hyperdocument model* is a generic document storage and retrieval model designed to provide an integrated framework for the management of standards provisions, their background information, program code representing the provisions (method objects), and external programs as a set of documents (Yabuki 1992; Yabuki and Law 1993). The model consists of a document base and a navigation system. The document base contains the electronic documents, where each document contains the content and a tag recording the information (such as keywords, rendering software etc.) about the document and pointers to referencing documents. The navigation system provides a set of generic facilities for document retrival: traversal of hyperlinks, query by keywords and pointers, and browsers.

To illustrate the hyperdocument model, a prototype system was implemented using Hypercard on a Macintosh computer. The hyperdocument system is linked and integrated with the object-logic system by sharing the method objects.

CD-ROM- and Internet-Based Approaches

Standards writing organizations are making their standards available through various CD-ROM and On-line services. For example, the following standards can be found on CD-ROM: All ASTM standards, the American Institute of Steel Construction Specifications and Manuals; the Americans with Disabilities Act; and all state and federal environmental regulations. Users of these standards have, in principle, immediate access to the most up-to-date versions of the text of these documents. Some of these systems allow only search over the links in the hypertext, while others offer the capability of full-text search over the entire text of the standards. However, full-text search over an entire standard, which is what most of the existing systems offer, is not very effective for locating the subset of applicable provisions within the standard of interest to the user.

To support standards users in locating and accessing applicable standards, Krofchik et al. have developed a standards broker that operates on the World Wide Web (Krofchik et al. 1995). The standards broker is a World Wide Web server from which distributed sources of information (in terms of both geographical and subject distribution) can be provided to users without the users having to know where that information resides on a wide area network. The standards broker makes use of the World Wide Web to display a hypertext form of the standard and permit the user to browse over the hypertext links; the search facilities provided in the broker allow the user to specify a design context and search for relevant specific provisions. This design context can consist of phrases to be used in full-text search and classifiers to be used in performing a classifier-based search.

ASSESSMENT OF PROGRESS

A Failure Analysis of Past Work

In the quarter century since the idea of a formal representation of design standards was first presented, and after a significant number of research papers on design standards processing have been published, only a handful of successful applications have been reported. What are the reasons for this significant mismatch between research ideas and practical applications?

The failure of the approach so far in the United States can be traced to two distinct areas. One major cluster of causes is methodological. Among the causes is, first, that the initial concepts were not backed up by usable, persistent computer tools. It was the intention of the SASE project to make such a collection of tools generally available, but this never became a reality. Second, there has been an explosion of research papers proposing new or extended formalisms, representations, concepts or approaches. As a consequence, concepts and terms kept changing, so that there has been no "equilibrium" state upon which tools and applications could be developed. While such innovative fervor is valuable, in this case methodological changes may have been counterproductive.

A related methodological shortcoming may be caused by a wrong choice of the initial application and of the initial model. Standards are generally classified as performance-based or prescriptive. The 1969 *AISC Specification* is neither; it may be called procedural: instead of emphasizing performance requirements, it emphasizes procedural steps to be performed to evaluate conformance with the requirements, which are often unstated (Fenves and Wright 1977). The *AISC LRFD Specification* is much more performance oriented, in that governing limit states are explicitly identified. Thus, the "decision table formulation" provided a model for representing the detailed procedural steps and checks for some, but by no means all, of the standards provisions, as described above for the ELRFD tool. The current modeling approaches have the promise of modeling the multitude of design standards that are not largely procedural. However, even these modeling approaches face major obstacles, as discussed in the next section.

The second cluster of causes of failure is professional, and has a lot to do with the dynamics of interaction of individuals and organizations. Individuals working essentially alone, or small, well-interacting groups working from within a standards writing organization can report successes, as in the case of the CSIRO BCAider project (Blackmore et al. 1991), Vanier working within the NRC of Canada (Vanier 1993), or Fenves working essentially on a one-to-one basis with the *AISC LRFD Specification* authors. When groups get larger, as design standards committees tend to be, the situation changes. The analyst, or small group of analysts, cannot keep up with the rest of the committee. Furthermore, such large groups tend to have an editorial subcommittee and that subcommittee may view itself solely responsible for the organization, format and textual expression of the standard. Such conflicts did arise in the case of the *ATC Provisions*. In addition, if the research group is not interacting with a standards writing organization, it does not have an opportunity to tune or illustrate its approach in realistic regulatory situations.

The two sets of causes, of course, interact. Without good tools to back up the analyst or, worse, tools directly usable by the design standards authors, there is not that much that the formalism can offer. The manager of a very large design standard development project responded to a summary of the methodology as follows: "There isn't anything in what you describe that a few smart people and a text processor couldn't do." Unfortunately, he was right.

Returning to the initial motivation of representing standards provisions for design standards processing, the methodology has also failed to make inroads in this area.

While there are only a few design standards represented with the concepts discussed, there is no pressing demand for computer-aided engineering systems that can process such representations.

FUTURE RESEARCH NEEDS

In the introduction, we stated that the initial research was motivated by the need to develop representations of standards on which computer tools could be developed. The kinds of computer tools envisioned were those for analyzing the standard itself and providing help to standards writers, those for assisting in finding applicable provisions of the standard, and those for performing automated conformance evaluation. Most of the previous work has focussed on design standard conformance evaluation, and even more specifically, on the representation and evaluation of provisions representable by first-order logic. While provision logic representation and computer-based conformance evaluation are the most actively researched aspects and are quite valuable, they are far from all that is needed by the authors and users of standards.

Based on our experience, more work is needed on the following aspects of standards representation and processing:

* representations of standards for supporting computation;
* computations on these representations;
* representations of standards for supporting access;
* assistance in accessing standards; and
* assistance in authoring standards and building models of standards.

The next five sections discuss the research needs in each of these five areas.

Representations for Computation

In order to better understand the methodological limitations of the standards modeling approaches discussed and to identify a broader range of standards modeling needs, several attempts have been made to use these modeling approaches on a variety of standards. The 1977 ACI Building Code Requirements for Reinforced Concrete (ACI 1977) were analyzed using SASE [Fenves et al. 87]. Garrett et al. have attempted to use several of the more recent modeling approaches described to model the following standards: Eurocode No.2: Design of Concrete Structures (EEU 1989); the 1993 BOCA National Building Code (BOCA 1993); and the Design Guide of the Department of the Army Office of the Chief Engineer DG-1110-3-145 (DAOCE 1986). While doing so, the need for modeling the following aspects of standards was encountered:

* design standards are not self-contained documents;
* design standards are indeterminate;
* processing design standards requires non-monotonic reasoning; and
* design standards contain higher order provisions;

Each of these obstacles is described in more detail in the following paragraphs.

Design standards are not self-contained documents. Most of the provisions of a design standard refer to knowledge that all humans are expected to know. Unfortunately this type of knowledge is not formally expressed anywhere. Everybody just knows it. Guha and Lenat refer to this type of knowledge as *pre-scientific* (Guha and Lenat 1990). In addition, understanding a design standard requires some knowledge about the domain of the design standard. Some domain knowledge, along with some *scientific* knowledge (the knowledge that engineers acquire in their education) is expected from the users of an engineering design standard. In addition, knowledge and heuristics are needed to know when to investigate another, referenced standard and when to proceed based on assumed compliance.

Design standard provisions are indeterminate. By indeterminacy, legal scholars mean the ability to argue either side of a question using accepted methods of legal discourse (Berman and Harfer 1988). Design standards, in that sense, are indeterminate. The most frequent cause of indeterminacy of provisions is caused by open-textured concepts used in expressing the provisions. An open-textured concept is one in which application to factual situations cannot be automatic, but which requires judgment and is context-dependent (Berman and Harfner 1988). Consider the following requirement from the 1993 BOCA National Building Code (BOCA 1993).

> **1017.4..4 Horizontal sliding doors:** [...] The door shall be openable from both sides *without special knowledge and effort* ;[...] (italics added)

The formalization of this provision requires the terms "openable without special knowledge" and "openable without special effort." Even if these terms are defined in great detail, there will always be cases that lie outside the definition. That means it is not possible to describe in advance which knowledge or effort is special.

To properly reason with the provisions of a standards, non-monotonic reasoning must be used. Non-monotonic reasoning deals with the derivation of plausible (but not infallible) conclusions from a knowledge base (Reiter 1987). Any such conclusion is understood to be tentative; it may be retracted at a later time due to additional information. Design standard processing requires non-monotonic reasoning because somtimes during processing some of the operations may have to be undone. For example, consider the following provisions from the 1993 BOCA National Building Code (BOCA 1993):

> **1006.6 Elevators, escalators and moving walks:** Elevators, escalators and moving walks *shall not be accepted* as a required element of the *means of egress*.
>
> **Exception [to 1006.6]:** An elevator conforming to Section 1007.3 *shall be permitted* for an accessible *means of egress*. (italics added)

In a monotonic reasoning system, if provision 1006.6 is evaluated and the elevators are judged to be unacceptable, it is not possible to undo this evaluation even if the elevator conforms to Section 1007.3. In this case, because the elevator conforming to Section 1007.3 is found to be acceptable according to the exception of the provision 1006.6, two conflicting results would be obtained.

Many provisions in standards are expressed in higher-order logic. About 40% of the provisions in the 1993 BOCA National Building Code (BOCA 1993) use second or higher order logic. First order predicate logic is the only logic used in current models of standards, but it is not sufficient to completely represent most of the current design standards. Although some provisions can be represented by reforming, or in some cases ignoring, the second order part, some of them cannot be expressed at all using first order logic. Consider the following provision from the 1993 BOCA National Building Code (BOCA 1993).

> **503.1 General:** The height and areas of all buildings and structures [...] shall not exceed the limitations fixed in Table 503, except as specifically modified by this chapter and the following Sections: 402.7, 403.3.3.1, 414.2, 416.3, 418.3.1.1, 3103.3.5

Because the provision references other provisions, it is not first order and thus can not be directly represented in current models of standards.

From this discussion, one obvious research need is to develop more powerful representations and processors of the information found in these design standards. As identified by various researchers (Guha and Lenat 1990; Johnson and Mead 1991; Sowa 1991) the development of powerful representations and processors is not simply a matter of scaling up from simple and experimental ones.

One possible solution to solve the scalability and other identified problems is to use a distributed framework to represent and reason with design standards. One immediate advantage of this solution is that it allows the incorporation of a wide variety of representational techniques into the same standard model, thus providing a broader, more powerful set of representations to use in modeling a design standard. Such a framework can also accommodate various types of knowledge bases and provide access to multiple design standards. Guides and heuristics supplied by the designers, companies or standard organizations may be modules in this distributed framework. These modules may be very complex and be running on multiple machines all over the world or be very simple and be running in the same address space on a single machine.

Computations

Along with more powerful representations for standards, a variety of different computational capabilities using these representations is needed. Users of standards must be supported in using standards at all stages of the design process. Today's standards processors basically only support the very detailed, post-synthesis phase of design, when a detailed description of the design is available and can be checked for conformance to a design standard. However, design standards are used by designers in the earlier stages of design, as well. For example, in the very early phases of design of a building, provisions from the local building code relating to the allowable area and height of the building must be taken into consideration. Thus, designers need processors that can help them find these applicable provisions for an ill-specified design context. Even for more detailed stages of the design process, the designer may want to derive guiding constraints from the design standard to use during the design process (similar to what SPEX did, but for much more complicated systems (Garrett and Fenves 1989)). Thus, a much wider variety of standard usages is needed than is currently supported by standards processors. Research is needed to clearly identify all of these different functionalities that are needed based on interactions with, and interviews of, actual users of design standards at all phases of design. Identification and delivery of this wider range of standard usage functionalities based on actual user studies might begin to address some of the reasons why standards organizations and users have not used this approach.

Representations for Access

Users of standards must be assisted in determining which standards, and which of the provisions in those standards, are applicable to his or her problem context. As we, Lopez et al., Vanier, and others have pointed out, determining which of the many standards, and the provisions in a standard, are applicable to a given design problem is a very important and difficult problem (Hakim and Garrett 1993; Lopez et al. 1989; Vanier 1993). In the SICAD prototype, Lopez et al. used a classifier tree-pruning technique to find applicable provisions (Lopez et al. 1989). Vanier uses a set of user-specified classifications and attribute values to eliminate provisions from a design standard as being inapplicable (Vanier 1993). In the prototype standards broker developed by Krofchik et al., users are able to find applicable standards and provisions for given problem classifications and textual phrases describing the problem context. Preliminary research into hypertext-based models of standards indicates that simply performing full-text search over a body of text will not greatly help a user to find applicable provisions due to the large amount of extraneous information that such searches produce (Krofchik et al. 1995). Reed (1991) and Krofchik et al. (1995) proposed breaking a standard down into its smallest clearly identifiable pieces — the provisions — and classifying each provision. Reed based his proposal on the original SASE approach to classification, but more research needs to be done regarding decomposition and classification strategies for these hypertext-based models. The classification of individual provisions is a vital ingredient to improving the search process for applicable provisions. What the facets of this classification system should be and whether they can be made detailed enough to classify each provision in a standard is a function of the problem domain to which the standard applies. This issue was studied by the developers of the SASE system. They developed a classification system, based on classification facets, that could be used to classify individual requirements in a design standard (Harris and Wright 1980). Such a faceted classification of the hypertext nodes in a hypertext-based models must be investigated. A related issue that needs to be investigated is the translation between the classification terms used by one individual standard and those used by other standards and standard users.

In both Vanier's and Krofchik's approaches, the user is forced to use the classifiers defined for a specific standard. In fact, most standards processing systems developed today assume that the user has an intimate understanding of the classes of objects referred to in the standard and expect their users to describe their problem in these terms. This assumption is a very difficult one to defend. As more and more standards are promulgated for specific issues, such as those related to the environment, users of standards will not be familiar enough with the terms used in a standard to properly use it. By the same token, the authors of the design standard should not be forced to use a general, standardized data model in building their model of a standard. Standard authors must be able to specialize this data model in order to facilitate the representation of the provision logic. What is needed is a general solution to the problem of mapping between different data schemas. The STEP activity may possibly lead to more generic classification structures to which the individual standard classifications can be related. In addition, the Construction Specification Institute has recently set in motion a series of events that may lead to the development of a new, more comprehensive classification system that applies throughout the life cycle of buildings, that is consistent with STEP and other information technology developments, and that simplifies this translational issue. Alternately, thesaurus creation and updating techniques need to be investigated. A third approach may be the object-centered approach described by Hakim (Hakim 1994). By intentionally describing classes, it is possible to determine if an object from another data schema is an instance of a class in another data schema. Obviously, this requires that the standard authors provide definitions for all of their terms and classes, but by doing so, they relieve the users from having to express their problems in terms defined only within a specific standard.

The hypertext and formal logic models for the most part have been researched and developed separately. Reed proposed a framework for the integration of these models, but did not implement it (Reed 1991). Law and Yabuki implemented an integration of these two models in the Hyperdocument Model (Yabuki and Law 1993b). More research in this direction is needed. It should be possible to start out using the hypertext model, with classified provisions, to identify the applicable provisions for a given design context and then move to the formal logic model of the standard to either compute conformance or derive constraints for use in design. It should also be possible to start out with a specific design context, use the formal logic model to identify and evaluate applicable provisions for a given design description, and then move into the hypertext model to investigate the definitions of the provisions in text form. This latter form of interaction was not supported by the Hyperdocument Model.

Access Assistance

Design standard users need assistance in discovering, selecting and accessing applicable standards. Design teams designing today's products are multi-disciplinary, geographically dispersed and multi-national. These design teams are expected to take more and more issues and objectives into consideration, many of which come about quickly but are none-the-less expected to be addressed by their designs. As these new design initiatives come about, they are accompanied by a large number of applicable publications, guidelines, specifications, and standards that need to be read, understood and integrated into the design process. Assuming that the designer, without assistance, is able to identify that a specific standard is applicable to his or her problem, quickly access that standard and incorporate it into the design process is unrealistic. However, almost all of the research being done in standards representation and processing assumes that the applicability of the standard being processed has already been determined and that the user has access to the most current formal or hypertext model of the design standard. More research must be done on how best to assist users in identifying and accessing formal and hypertext models of applicable standards.

Krofchik et al. have started to investigate the role of the Internet in providing the distributed delivery of and access to standards. In this approach, both the standard agencies and the standard users are distributed (geographically and organizationally); the standard agencies "serve" their most up-to-date models of their standard to users describing design contexts to which the standard is applicable. Many enabling technologies for serving documents, searching for documents, billing for document

Fenves, S. J., E. H. Gaylord, and S. K. Goel, 1969, *Decision Table Formulation of the 1969 AISC Specification,* Civ. Eng. Std. SRS 347, Deptartment of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.

Fenves, S. J., K. Rankin, and H. K. Tejuja, 1976, *The Structure of Building Specifications,* NBS BSS 90, NBS, Washington, DC.

Fenves, S. J. and R. N. Wright, 1977, *The Representation and Use of Design Specifications,* Technical Note 940, NBS, Washington, DC.

Fenves, S. J., R. N. Wright, F. I. Stahl, and K. A. Reed, 1987, *Introduction to SASE: Standards Analysis, Synthesis and Expression,* NBSIR 873513, NBS, Washington, D.C.

Garrett, Jr., J. H. and S. J. Fenves, 1986, *A Knowledge-Based Structural Component Design*, Report No, R-86-157, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA.

Garrett, J. H., Jr., and S. J. Fenves, 1987, A Knowledge-based standard processor for structural component design, *Engineering with Computers,* **2**(4), pp 219-238.

Garrett, Jr., J. H. and M. Maher Hakim, 1992, An object-oriented model of engineering design standards, *Journal of Computing in Civil Engineering,* **6**(3), pp 323-347.

Goldfarb, C. F., 1990, *The SGML Handbook*, Oxford University Press.

Guha R. V., and D. B. Lenat, 1990, Cyc: A Mid-Term Report, *AI magazine*, **11**(3), pp 31-59.

Hakim, M., M. and J.H. Garrett, Jr, 1992, A Design Component Library Based on Design Standards, *Proceedings of the Eighth Conference on Computing in Civil Engineering,* ASCE, pp 105-112.

Hakim, M. M., 1993, *Modeling evolving information about engineering design products,* Ph.D. dissertation, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh, PA.

Hakim, M. M. and J. H. Garrett, Jr, 1993, A Description Logic Approach for Representing Engineering Design Standards, *Engineering with Computers*, **9**, pp 108-124.

Harris, J. R., S. J. Fenves, and R. N. Wright, 1979, *Analysis of tentative seismic design provisions of buildings,* Tech. Note 1100, NBS, Washington, DC.

Harris, J. R. and R. N. Wright, 1980, *Organization of Building Standards: Systematic Techniques for Scope and Arrangement*, NBS BSS 136, NBS, Washington, DC.

International Organization for Standardization (ISO), 1986, *ISO 8879: Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)*, ISO Central Secretariat, Geneva, Switzerland.

International Organization for Standardization/The International Electrotechnical Commission (ISO/IEC), 1992, *ISO/IEC 10744: Information Technology -- Hypermedia/Time-based Structuring Language*, ISO/IEC Central Secretariat, Geneva, Switzerland.

Newcomb, S. R., N. A. Kipp, and V. T. Newcomb, 1991, The HyTime Hypermedia Time-Based Document Structuring Language, *Communications of the ACM*, **34**(11) pp 67-83.

Noland, J. L., and C.C. Feng, 1975, ACI Building Code in Decision Logic Table Format. *ASCE Journal of the Structural Division*, **101**(4), pp 677-696.

Nyman, D. J. , S. J. Fenves and R. N. Wright, 1973, *Restructuring Study of the AISC Specification*, Civil Engineering Standards, SRS 393, Department of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.

Nyman, D. J., and S. J. Fenves, 1975, An Organization Model for Design Specifications. *ASCE Journal of the Structural Division*, **101**(4), pp 697-716.

Rasdorf, W. J. and T. E. Wang, 1988, Generic design standards processing in an expert system environment, *Journal of Computing in Civil Engineering*, **2**(1), pp 68-87.

Rasdorf, W. J. and S. Lakmazaheri 1990, Logic-based approach for modeling organization of design standards, *Journal of Computing in Civil Engineering*, **4**(2), pp 102-123.

Rasdorf, W. J. and S. Lakmazaheri, 1990, A Logic Based Approach for Processing Design Standards, *International Journal of Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, **4**(3), pp 179-192.

Reed, K. A., 1991, Building Regulations and Standards in the Computer Age, *Computers and Building Regulations*, K. Kahonen and B-C. Bjork, eds., Technical research Center of Finland (VTT).

Rehak, D. R. and L. A. Lopez, 1981, *Computer Aided Engineering Problems and Prospects*, Civil Engineering Standards SRS 8, Dept. of Civil Engineering, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.

Reiter, R. 1987, Non-monotonic reasoning, *Annual Review of Computer Science*, **2**, pp 147-186.

Sowa J. F, 1991, Current Issues in Semantic Networks, *Principles of semantic networks*, J. F. Sowa, ed. San Mateo CA: Morgan Kauffmann, pp 13-43.

Topping, B. H. V., and B. Kumar, 1989, Knowledge representation and processing for structural design codes, *Engineering Applications of AI*, **2**(9), pp 214-227.

Vanier, D., 1993, Minicode Generator: A Methodology to Extract Generic Building Codes, *Proc. of CAAD Futures '93*, ed. U. Flemming and Van Wyk, North-Holland Publishers.

Yabuki, N., 1992, *An Integrated Framework for Design Standards Processing*, Ph.D. dissertation, Department of Civil Engineering, Stanford University, Palo Alto, CA.

Yabuki, N. and Law, K.H, 1993, An Object-Logic Model for the Representation and Processing of Design Standards, *Engineering with Computers*, **9**, pp 133-159.

Yabuki, N. and K. H. Law, 1993b, HyperDocument Model for Design Standards Documentation, *Journal of Computing in Civil Engineering*, ASCE, **7**(2), pp 218-237.

International Organization for Standardization (ISO), 1995, *ISO 10303: Industrial Automation Systems and Integration --Product Data Representation and Exchange*, ISO Central Secretariat, Geneva, Switzerland. This standard is being issued in parts as they are approved.

Jain, D., K. H. Law and H. Krawinkler, 1989, On processing standards with predicate calculus, *Proc. of the Sixth Conference on Computing in Civil Engineering,* ASCE, T. Barnwell, ed., Atlanta, GA, pp 259-266.

Johnson, P., and D. Mead, 1991, Legislative knowledge base systems for public administration-some practical issues, *Proceedings of the Third International Conference on Artificial Intelligence and Law*, Oxford, UK, 25-28 June, pp 108-17.

Kiliccote H, 1994, *The context-oriented model: a hybrid approach to modeling and processing design standards*, Master's thesis, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA.

Kowalski R., and M. Sergot, 1991, The use of logical models in legal problem solving, In *Law, computer science, and artificial intelligence*, ed. A. Narayanan and M. Bennun, pp 99-118. Norwood, N.J. : Ablex Publication Corporation.

Lakmazaheri, S., 1990, *A Study on the Constraint Logic Approach for Structural Design Automation*, Ph.D. dissertation, Department of Civil Engineering, North Carolina State University, Raleigh, NC.

Lauritsen M., 1991, Knowledge-based approaches to government benefits analysis, In *Proceedings of the Third International Conference on Artificial Intelligence and Law,* Oxford, UK, 25-28 June 1991, pp 98-107.

Law, K. H. and N. Yabuki, 1992, An integrated system for design standards processing, In *Proceedings of the 1992 Computer and Bldg, Standards Workshop*, ed. D. Vanier, Univ. of Montreal.

Lopez, L. A., and S. L. Elam, 1984, *SICAD: A Prototype Knowledge Based System for Conformance Checking and Design,*Technical Report, Department of Civil Engineering. University of Illinois at Urbana-Champaign, Urbana-Champaign, IL.

Lopez, L. A., and R. N. Wright, 1985, *Mapping Principles for the Standards Interface for Computer Aided Design,* NBSIR 85-3115, National Bureau of Standards, Gaithersburg, MD.

Lopez, L. A., S. Elam and K. Reed, 1989, Software concept for checking engineering designs for conformance with codes and standards, *Engineering with Computers,* 5:63-78.

MacKellar, B. K. and F. Ozel, 1991, ArchObjects: Design codes as constraints in an object-oriented KBMS, *Proc. of the First International Conference on Artificial Intelligence in Design,* Edinburgh, UK.

Malasri, S., Olabe, J.C. and M.A. Olabe, 1991, A Prototype of the Uniform Building Code Using Hypertext, *Proceedings of the 10th Conference on Electronic Computation,* ASCE, pp 36-43.