6. An Integrated Model for the Representation, Processing, and Documentation of Design Standards

Nobuyoshi Yabuki [1] Kincho H. Law [2]

【Abstract】 A formal and theoretical model for representing design standards is required in order to automate design and to develop design software efficiently. We developed a Hyper-Object-Logic Model, which is an integration of an Object-Logic Model, which object-oriented and logic programming paradigms are unified and HyperDocument Model for documentation of design standards and their related information. By using this model, one can systematically develop design software, linking provisions of design codes and their programs. This model can be used for conformance checking of design and automated design generation. This paper also discusses a prototype system and future research directions.

【Keywords】 design standards, design automation system, object-oriented programming, logic programming, information sharing

1. Introduction

Since the dawn of computers, they have been extensively used in civil engineering for improving efficiency and advancing design. For example, generally, the order of design task of structures is structural analysis, design calculation, and drafting. Finite element analysis and CAD systems are widely used in structural analysis and drafting respectively. And general-purpose codes and commercial CAD software are extensively used. On the other hand, design calculation based on design standards may not be as computerized as structural analysis and drafting. Thus, computers are not effectively utilized in the design process in an integrated manner.

A design standard is a document that states the requirements that must be met in order to ensure that an adequate level of performance for a structure or members. Design standards play a significant role in the design process to ensure safety, quality, and functionality of civil engineering structures.

Various programs have been developed for processing design standards in order to support design process. However, most of the programs are written in procedural languages such as FORTRAN and they are developed for either conformance checking or automated design generation but not for both of them. It is difficult to modify the program when the standard is revised because the provisions are "hard-coded" into the program. Hard-coded programs often aim for either conformance checking or automated design, but it is rather difficult to implement both applications in a unified manner. Current conformance checking programs are mostly stand-alone type and are not integrated with CAD data models, engineering databases, computer aided drafting systems. The user of such a conformance checking system must supply to the system all the necessary input data by hand. Once the user starts to use such a system, the user tends to design by the system without fully understanding provisions and commentary.

On the other hand, although the

1. Associate Professor, Dept of Civil Engineering and Architecture, Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran-shi, 050-8585, TEL: 0143−46−5219, FAX: 0143−46−5218, Email: yabuki@news3.ce.muroran-it.ac.jp
2. Professor, Dept. of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305-4020, U.S.A.,

organization concept and writing style of provisions are consistent within each design standard, they are usually not based on a formal model and are arbitrary. The user of design standards must learn how to use them for each standard, which also hinders development of conformance checking programs.

For more than thirty years, many research projects on developing formal models for computerizing design standards organization and provisions have been executed and many research results have been generated. However, few models can be evaluated as universal or general.

In this paper, we describe an integrated and theoretical model for design standards representation, processing, and documentation, and address future issued and directions of research.

2. Previous Research

The research on design standards organization and provision processing is divided into the development of formal models and computerization of design standards as documents. Their outline and evaluation are discussed in this section.

(1) Formal Models

The first formal model for design standards processing[1] was based on decision tables and has been modified for a number of times and it is called Standards Analysis, Synthesis, and Expression (SASE) model[2].

The SASE model consists of four basic components:

(i) data items, which represent all the variables in the standard,
(ii) decision tables, which represent the logic used to determine the values of data items,
(iii) information networks, which represent the precedence relations among the data items, and
(iv) an organization system, which represents the organization of the provisions.

Subsequently, a design standards processing methodology, which provisions are represented as production rules and which design is verified by an inference engine was proposed[3]. Then, a frame-based and object-oriented approach was proposed for representing the organization and data items respectively[4]. Furthermore, predicate logic, which is one of the methodologies of knowledge representation in artificial intelligence, was applied for design standards and the possibility of design verification and partially automated design generation was demonstrated[5].

Later on, we have thought that object-oriented programming is natural for representing design objects organization and that predicate logic is suitable for representing and processing provisions and have developed Object-Logic model[6] by merging these two paradigms.

(2) Computerizing Design Standards as Documents

Provisions often refer to other provisions in design standards. Thus, hypertext, which references can be embedded in a document as links, has been used to computerize design standards and stored in floppy disks and CD-ROMs[7]. However, those systems are limited to computerize existing design standards as they are and a formal model for design standards has not been addressed yet.

In the process of design standards development, a large amount of related documents and data are collected by design standards committee members. However, since related documents are too large to be included in design standards, this kind of deep knowledge cannot be shared by engineers and designers, who are the users of the design standards. Thus, we have developed HyperDocument model[8] for linking the background knowledge and design standards and for supporting design task. This model is based on HyperFile[9] and is a general model that can store documents and files as well as computer programs representing design standards and related computer programs.

(3) Evaluation of Previous Research

For more than thirty years, in this research field, many research papers have been published, especially in the United States, and various prototype systems, which are mostly developed for processing American Institute of Steel Construction (AISC) – Load and Resistance Factor Design (LRFD) specification[10). However, there have been few cases that those research models are realized for practical use. The reason why they are not realized like general-purpose codes for finite element methods (FEM) and commercial CAD systems for drafting is considered. FEM codes developed in university laboratories could be modified to become commercial codes that could be used by practical engineers in the real world with relatively small amount of extra work. As for the CAD system, although substantial amount of software development was necessary in order to change a prototype system to a commercial one, it was obvious that the benefit was expected to be larger than the additional software development cost.

On the other hand, design process by using design standards is an essential task for design engineers and it takes long time and experience for them to acquire such competence. Thus, the design process is difficult for software engineers who are not accustomed to design, and it is hard for them to compare the benefit and cost. Moreover, as new formal models have been developed and introduced one after another in the academic field, it is natural for software industry people have decided to wait until the model becomes mature enough.

Although design standards committee members realize the importance of computerizing design standards, they may not familiar with formal models of design standards and due to the limited time, effort, and budget, they may think that some other organization or software companies should execute the computerization task.

Currently, there is a trend and effort of transition from procedural design to performance-based design in order to provide designers and engineers with more flexibility and freedom of choice based on their engineering knowledge and judgment. However, guidelines and manuals for basic design are still necessary. Especially, small to medium-sized structures should be designed efficiently. Design standards play an important role as a risk mitigation measure when a failure or damage occurs due to some reason. So, provisions of design standards should be carefully written in a manner so that new design methodologies or technologies are not to be excluded or impeded if safety and functionality are ensured by other more advanced methodologies. And, in order for design engineers to have high level engineering judgment competence, more time to think should be generated by computerizing cumbersome task of design based on design standards.

3. Hyper-Object-Logic Model

In the previous research, formal model development and computerization of design standards as documents have been done separately. So, we have integrated Object-Logic model and HyperDocument model to develop Hyper-Object-Logic model that can:

- represent both the organization and provisions of the design standard effectively,
- perform both conformance checking and component design within the same environment,
- store background documents and knowledge that can be accessed by code writers and engineers.

A prototype system has been developed for AISC-LRFD specification[11) on an Apple Macintosh personal computer by using Prolog++[12), HyperCard, Oracle database system, etc. The organization of the model is shown in **Figure 1**, and components of the model are described hereafter with application examples of the prototype system.

(1) Standards Base

As a formal model of design standards, an organizational model that all the applicable provisions can be accessed and that can cover all the provisions without any anomalies such as lack of completeness, redundancy, and contradiction, and a model that can represent the logic of provisions so that conformance checking of design can be executed are necessary. That is, the model should be used to reorganize the structure and provisions instead of writing programs from the design standard as it is.

In the Object-Logic model, the organization of design standard is constructed by merging three tree-structure fields of members, stress states, and limit states based on the SASE model and is represented as an object-oriented standards organization hierarchical tree. Each

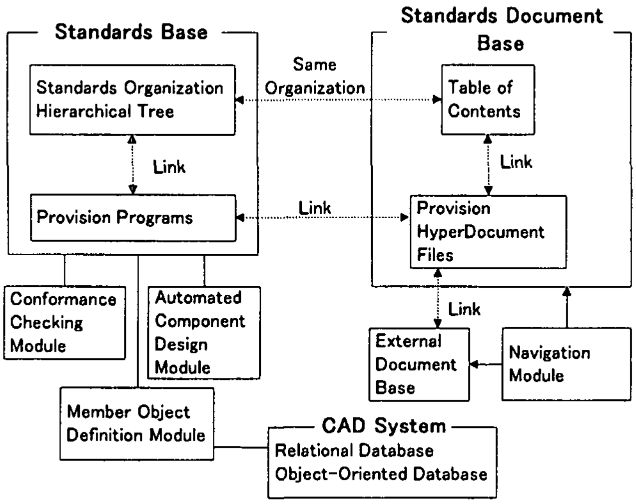

Figure 1 Hyper-Object-Logic Model

requirement is attached to a corresponding leaf mode of the tree. An example is shown in **Figure 2**. This organization enables checking the lack of completeness and redundancy at each level of the hierarchy and accessing applicable requirements by traversing the tree from the root to lowest leaf nodes if a member and its attribute values are given.

Each node of the standards organization hierarchical tree is classified as either AND node or OR node. When traversing the tree, all nodes under an AND node are traversed, while only one node under OR node is selected. A part of the standard organization hierarchical tree is shown in **Figure 3**. A requirement "req_short_non_slender_y_comp_mem" contains a condition that must be satisfied for a steel compression member with non-slender sections and limit state of weak axis inelastic buckling. In the figure, arrow lines indicate traversal by the system for a given member named "column #247."

Programs representing provisions are described as method objects. Each method object corresponds to a single data item, and contains a method that computes the data item. Method objects are classified as requirements; determinants which are needed to determine other data items; and classifications which classify each member. Methods are written in Object-Logic sentences expressed as:

A :- $C_1, C_2, ..., C_n$. or A.

A is a conclusion written in the form *Method(Term*$_1$*, Term*$_2$*, ..., Term*$_m$*)*

where C_i (i = 1, 2, ..., n) is a condition which has the following structure:

Mem::Method<-Message, *self<-Message* or an arithmetic expression.

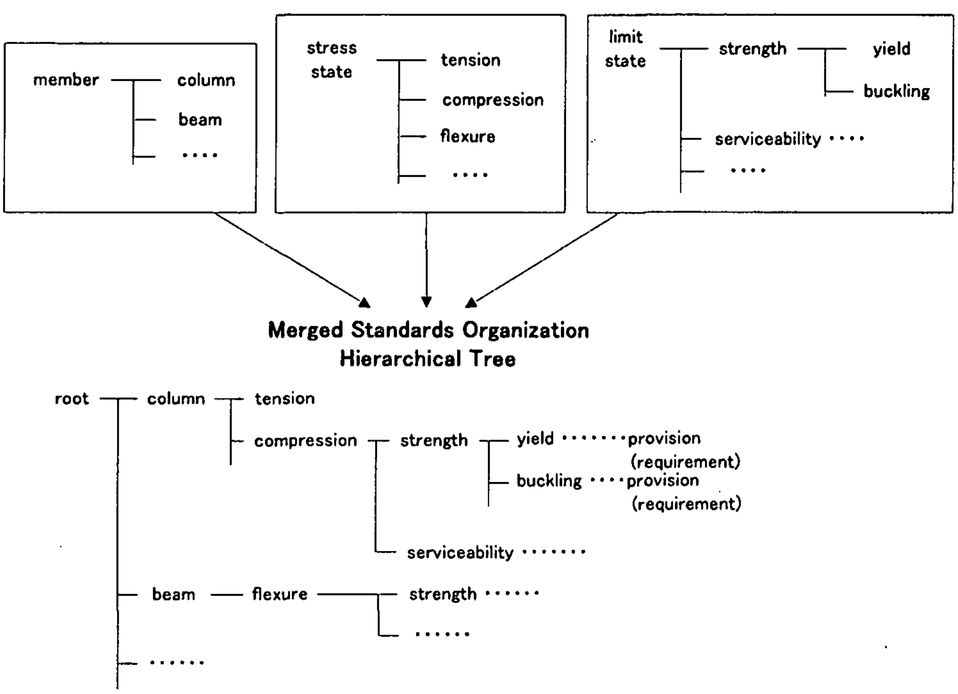

Figure 2 A Simple Example of Development of Standards Organization Hierarchical Tree

Term is a variable, object constant, or functional expression. *Message* is a literal that has the same form as the conclusion *A*. The symbol, :-, means that if a right hand side is true the left hand side is true. The symbol, <-, means that the right hand side *Message* is sent to *Method* or *self*.

When a design member is given, the resolution principle works as an inference engine and generates verification results. As an example, requirement for a weak axis inelastic buckling is shown in **Figure 4**.

(2) Member Object Definition Module

A member object such as a beam or column consists of attributes including physical properties and dimensions, and external constraints. In this model, a member object is defined as an instance of a leaf node class in the hierarchical tree which consists of steel columns, steel beams, reinforced concrete (RC) beams, etc. By the object-oriented programming paradigm, attributes in upper classes are inherited to lower classes and additional attributes can be added as necessary. Attributes of a column named "column_25" (**Figure 5**) in a structure are shown in **Figure 6**.

In a CAD environment, attribute data of design member objects can be stored in a relational or object-oriented database that is linked to 3D CAD system. Attribute data can be automatically retrieved from the database via an appropriate interface.

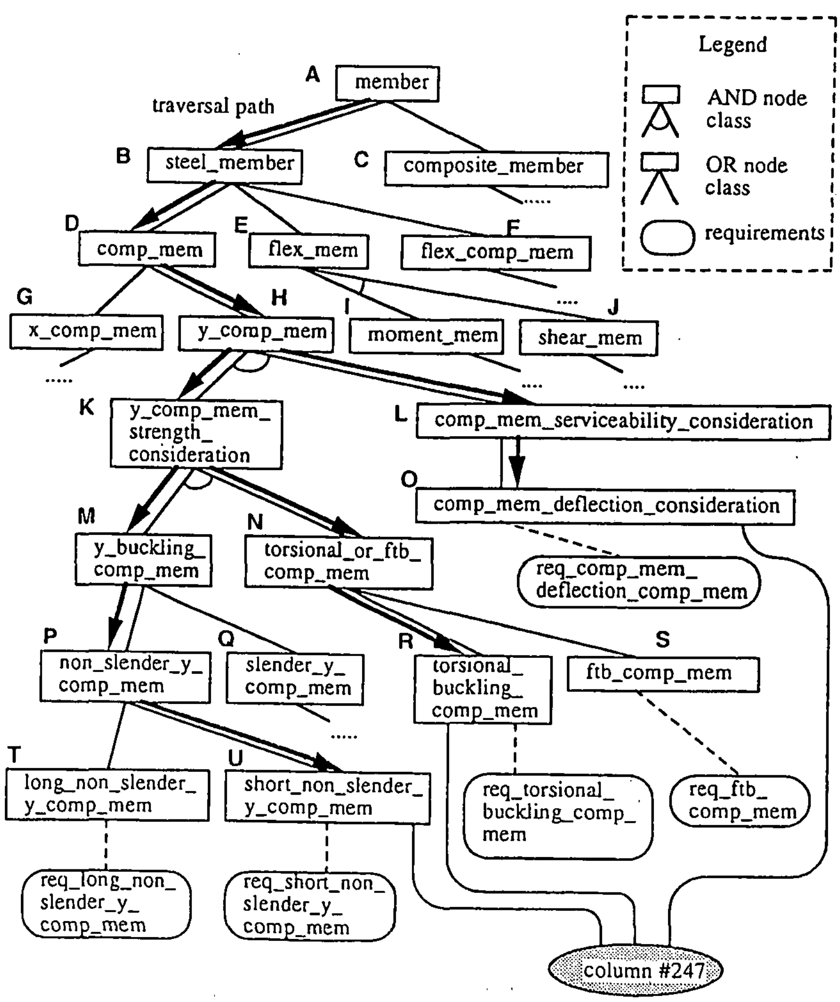

Figure 3 A Part of the Standards Organization Hierarchical Tree

```
open_object req_short_non_slender_y_comp_mem.

  super = requirements.
  class = short_non_slender_y_comp_mem.
  provision = 'E2'.
  reference = [pn_short_y,
               attr(load_comp)].
  meaning = 'requirement for a non-slender compression
             member which y-axis inelastic buckling governs'.

  req_short_non_slender_y_comp_mem(Mem,satisfied,Id):-
      Mem::pn_short_y <- pn_short_y(Mem,Pn,Id),
      DS is 0.85 * Pn,
      self <- input_attr(Mem,load_comp,1000),
      Mem::eqless <- eqless(Mem::load_comp,DS),
      self <- respond_satisfied.

  req_short_non_slender_y_comp_mem(Mem,violated,Id):-
      Mem::pn_short_y <- pn_short_y(Mem,Pn,Id),
      DS is 0.85 * Pn,
      not(Mem::eqless <- eqless(Mem::load_comp,DS)),
      self <- respond_violated.

close_object req_short_non_slender_y_comp_mem.
```

Figure 4 An Example of Requirement Method Object

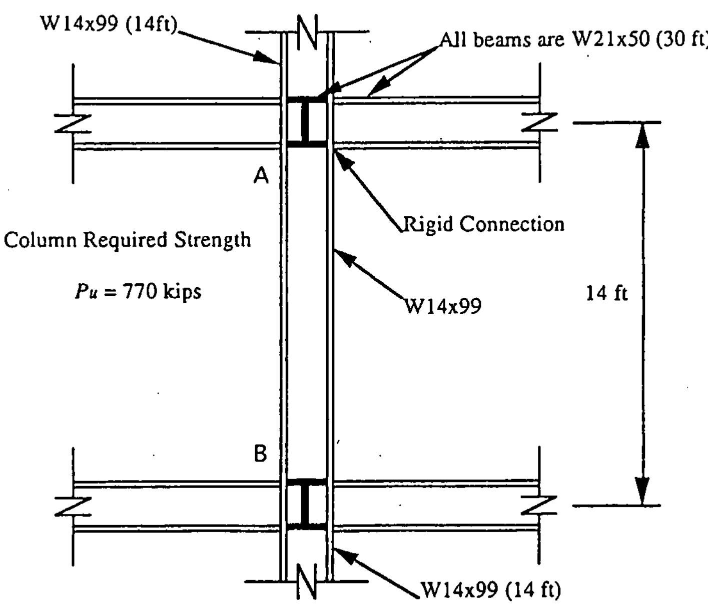

Figure 5 Column_25

Defining Compression Member
Member Name : column_25

Attribute	Value	Default	Unit
elastic_modulus	is 29000.	% 29000	ksi
shear_modulus	is 11000.	% 11000	ksi
yield_stress	is 36.	% 36	ksi
unbraced_length_x	is 168.	% 120	in
unbraced_length_y	is 168.	% 120	in
effective_length_factor_x	is unknown.	% 1	
effective_length_factor_y	is unknown.	% 1	
shape	= wshape.		
load_comp	is 720.	% 300	kips

Figure 6 Attribute Data of Column_25

(3) Conformance Checking Module

Conformance checking refers to the evaluation of a design member as to whether it satisfies the applicable requirements as defined in a design standard. In this module, first, member object definition module is executed, and then the system traverses the standards organization hierarchical tree and all the applicable requirements are selected and executed. Then, the result is shown to the user in the form of design verification report.

For the compression member "column_25" example, one of four applicable requirements is violated and others are satisfied as shown in **Figure 7**.

(4) Automated Component Design Module

Since design has more than one solution and is often an ill-defined problem, a trial-and-error method is usually used in order to reach an optimized design. In this model, the design approach is to generate a plausible component design using a set of heuristics and to test the design for conformance checking. The design heuristics are procedures applicable for a group of members governed by appropriate requirements. The knowledge and peripheral processing functions are clearly separated from the design standards contents and are developed as a separate module. This module is written in Object-Logic sentences.

(5) Standards Document Base

Each provision as well as figure, table, chart is stored as HyperDocument file. References among provisions are represented as hyperlinks and a provision and its corresponding program are hyperlinked together. By these links, when a design violates some requirement, the user can read the corresponding provision by clicking the button shown in **Figures 7** and **Figure 8**.

The standards organization hierarchical tree has the same structure and organization as the table of contents of the standards document base in order to integrate these two bases.

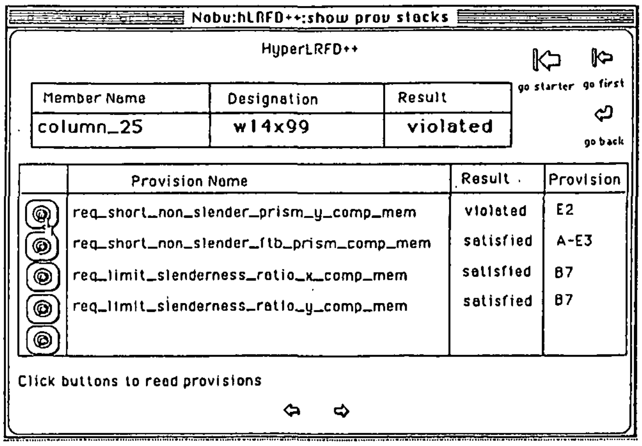

Figure 7 Verification Result of Column_25

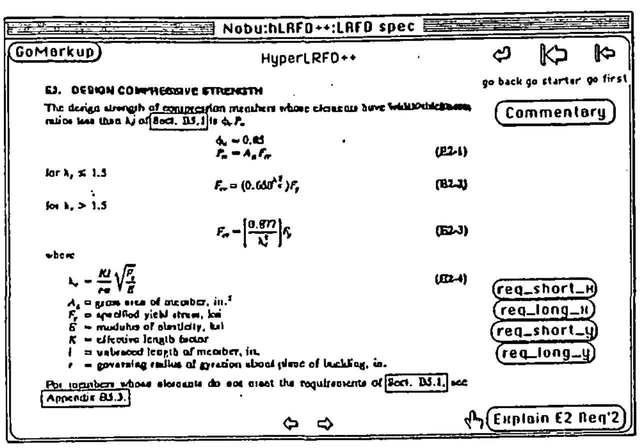

Figure 8 Violated Provision E8

(6) External Document Base

A large amount of background information behind the standard such as commentaries of provisions, other design standards referenced by the design standards, and related documents are stored in a computer. The user can easily access and read such documents from the standards document base. For example, consider a case that the user intends to verify the design of a column and to obtain the effective length factor, K, to determine the effective length of the column. The AISC-LRFD specification contains an alignment chart to evaluate the K factor from restraint factors for the end joints. However, the specification suggests the user should refer to other papers when inelastic buckling governs. In this case, the user clicks the button indicating the referenced document[13] in the commentary and the scanned paper is displayed on the computer screen easily as shown in **Figure 9**.

(7) Navigation Module

The navigation module allows the user to retrieve documents from the document base easily and efficiently. Hyperlinks, keywords, and browsers are provided to facilitate retrieving documents as shown in **Figure 10**.

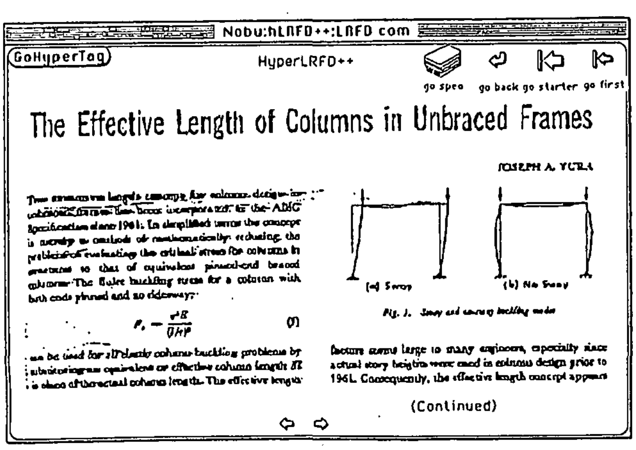

Figure 9 Referenced Paper from a Commentary of AISC-LRFD Specification

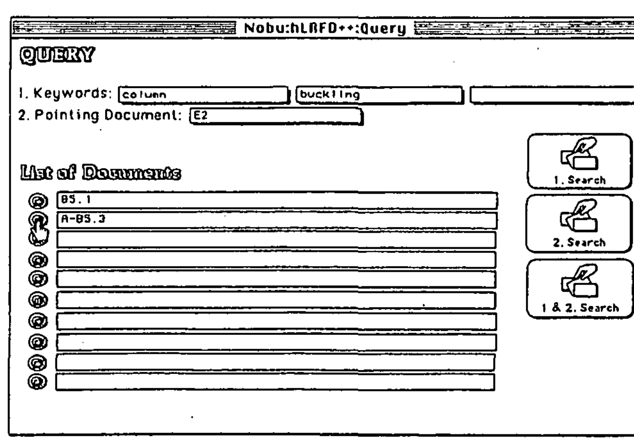

Figure 10 Query Page and a List of Referenced Documents

4. Summary and Discussion

In this paper, we reviewed and evaluated the previous research on various formal models for representing and processing design standards, and described Hyper-Object-Logic model and its prototype system. Although this model demonstrated its applicability and feasibility, limitations of this research and directions for future work are discussed in this section.

A design standard does not exist in a closed world but it references to other design standards and retrieves data from them or executes numerical processing. Referenced standards may reference to other documents. In order for various design standards processing system communicate smoothly and without any error, a standardized rule such as CORBA (Common Object Request Broker Architecture) may be necessary.

Furthermore, it is an issue how to deal with the situation that a program of referenced standards is modified and their addresses in the network or attributes of classes are changed. To solve such a problem, an agent that notifies the user various changes made in the referenced systems by moving in the network.

The prototype system discussed in this paper was developed on a Macintosh computer and is being transferred to and modified on a Windows personal computer. In the current development, HyperText Markup Language (HTML) and Extensible Markup Language (XML) are employed.

Characteristics of design standards vary dependent on nations, structures, scopes, etc. The applicability of the model should be tested for design standards containing ambiguous provisions. In this research we used general Object-Logic sentences for developing the prototype system. However, some specific and flexible language may be necessary in order to develop programs efficiently.

Lastly, in this model the result of design verification is satisfied or violated. But ambiguous evaluations such as favorable and undesirable, and philosophical situation such as sublation should be addressed in the future work.

References

1) Fenves, S. J.: Tabular Decision Logic for Structural Design, *Journal of the Structural Division*, Proc. of ASCE, Vol. 92, No. ST6, pp. 473-490, 1966.

2) Fenves, S. J., Wright, R. N., Stahl, F. I., and Reed, K. A.: Introduction to SASE : Standards Analysis, Synthesis, and Expression, National Bureau of Standards, Report No. NBSIR 87-3513, 1987.

3) Rosenman, M.A. and Gero, J.S.: Design Codes as Expert Systems, *Computer-Aided Design*, Vol. 17, No. 9, pp. 399-409, 1985.

4) Garrett, Jr., J. H. and Fenves, S. J.: A Knowledge-Based Structural Component Design, Report No. R-86-157, Dept. of Civil Engineering, Carnegie-Mellon University, 1986.

5) Rasdorf, W. J. and Lakmazaheri, S.: A Logic Based Approach for Processing Design Standards, *International Journal of Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 4, No. 3, pp. 179-192, 1990.

6) Yabuki, N. and Law, K. H.: An Object-Logic Model for the Representation and Processing of Design Standards, *Engineering with Computers*, Springer-Verlag, Vol. 9, No. 9, pp. 133-159, 1993.

7) Cornick, S. M.: HyperCode: The Building Code as a Hyperdocument, *Engineering with Computers*, Vol. 7, No. 1, 1991.

8) Yabuki, N. and Law, K. H.: HyperDocument Model for Design Standards Documentation, *Journal of Computing in Civil Engineering*, ASCE, Vol. 7, No. 2, pp. 218-237, 1993.

9) Clifton, C. and Garcia-Molina, H.: Distributed Processing of Filtering Queries in HyperFile, Technical Report No. CS-TR-295-90, Dept. of Computer Science, Princeton University, 1990.

10) Yabuki, N.: An Integrated Framework for Design Standards Processing, Ph.D. Thesis, Dept. of Civil Engineering, Stanford University, 1992.

11) Manual of Steel Construction – Load & Resistance Factor Design, Second Ed., American Institute of Steel Construction, Inc., 1993.

12) Moss, R.: Prolog++: The Power of Object-Oriented and Logic Programming, Addison-Wesley, 1994.

13) Yura, J. A.: The Effective Length of Columns in Unbraced Frames, *Engineering Journal*, Vol. 8, No. 2, pp. 37-42, 1971.