NRC Publications Archive Archives des publications du CNRC

Hypercode : The Building Code as a hyperdocument Cornick, S. M.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Engineering with Computers, 7, 1, pp. 37-46, 1991

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=7722287a-672e-4289-94d4-a58efd339600>
<https://publications-cnrc.canada.ca/fra/voir/objet/?id=7722287a-672e-4289-94d4-a58efd339600>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

HyperCode: The Building Code as a Hyperdocument

S.M. Cornick

National Research Council of Canada, Institute for Research in Construction, Advanced Construction Technology Laboratory, Calgary Alberta, Canada

Abstract. The Canadian National Building Code serves as a model document for several other legal documents that govern the construction of buildings, and as such it consists primarily of a list of definitions, prescriptive rules, and performance requirements, and contains cross-references to other legal and quasi-legal documents. There are many different users of this document, each using the Code for its own ends. There is also a considerable amount of variance among the individual user groups. The concept of hypertext, developed in late 1930s and mid-1940s, exhibits characteristics that make it an ideal medium for the development of an electronic Code. The National Building Code conceived as a hyperdocument would allow for much greater flexibility in accessing the information contained therein. A HyperCode would also enable users to tailor the document to their specific needs.

1 The National Building Code of Canada

The National Building Code of Canada (the Code or the Building Code) [1] is a model document for several other legal documents that govern the construction of buildings. It is essentially a set of minimum provisions respecting the safety of buildings with reference to public health, fire protection, and structural sufficiency. Content of the Code is determined by several standing committees that carefully examine the content, intent, and wording of each piece of information. The Code is published every 5 years, and revisions to the current edition are published on a yearly basis. This continual updating reflects the impact of social and technological changes on buildings.

In general terms the Code defines objects, such as walls, doors, beams and so on, and the relationships and constraints that exist among these objects. Technical requirements in the Code serve to constrain the attributes of objects, for example, the composition of firewalls, as well as the relationships

among objects, such as the distance that combustible projections are located from one another. The document itself is a large collection of definitions, prescriptive rules, performance specifications, and cross-references to other documents.

The Building Code is structured hierarchically and consists of several parts, each having a defined extent and scope. Each part is divided into sections and subsections, each of which deals with specific subtopics in the hierarchy. This type of subdivision can continue for several more levels (see Fig. 1.).

Most of the information in the Building Code is indexed using a seven-character code. This encoding allows for the generation of links across the hierarchy enabling a reader to traverse the tree horizontally. A sample of the Code is shown here:

3.1.8.1.(13) Where the external walls of 2 buildings meet a firewall at an angle of 135° or less, the requirements of Article 3.2.3.10. shall apply.

The excerpt is from Part 3, Section 1, Subsection 8, Article 1, Sentence 13, and refers across the hierarchy to Section 2 of Part 3. The italicized words in the excerpt are another form of link found in the document. Italicized words have a precise meaning within the Code and are defined in a section of the document akin to a glossary. A variety of additional pictorial and textual information is provided within an appendix. This additional information is not structured, but is loosely associated with the main body of the document (i.e., a return link is provided). Some of the individual parts of the Code also refer to supplements, which contain voluminous data tables, such as climatic information as well commentaries on the document's contents. References to several other codes of practice and standards can also be found in the Building Code. In fact, the Code refers to a total of 192 other documents.

The current version of the Code contains annotations within the margins that provide a brief general description of segments of the document. These an-

Offprint requests: S.M. Cornick, National Research Council of Canada, Institute for Research in Construction, Advanced Construction Technology Laboratory, 6815 8th Street N.E. Calgary Alberta, Canada, T2E-7H7 (e-mail: cornick @ noah. arc. ab. ca).

3	Part
3.5	Section
3.5.1	Subsection
3.5.1.6	Article
3.5.1.6.(1)	Sentence
3.5.1.6.(1).(e)	Clause
3.5.1.6.(1).(e).(i)	Subclause

Fig. 1. Code document hierarchy.

notations are to be replaced with article titles in future editions. In an attempt to draw the reader's attention to changes from the current edition, a vertical bar is placed alongside new information. Vertical bars serve to link the current edition to previous editions, establishing a temporal link between versions of the document (see Fig. 2).

These two features of the Code, annotations and change bars, serve to highlight the issues of revisions and version control that are discussed in a later section of this paper.

In summary, the National Building Code of Canada forms a highly complex document, governing the construction of buildings. A considerable amount of structuring has been imposed on the document; however, it still remains a rather difficult document to use.

2 The Users of the Code

There are four principal groups of users of the Building Code: (1) provincial and municipal governments, (2) municipal building inspectors, (3) architects, and (4) engineers. Each of these groups tends to view and utilize the Code in a unique fashion.

Governments use the Building Code as a basis for legal documents within their jurisdiction, and as such the Code is viewed as a model document. Changes to various portions of the Code are made

to reflect social, climatic, and technological factors that are peculiar to the region where it is to be applied. Some governments make substantial changes to the document, whereas others make no changes whatsoever.

Building inspectors have the responsibility of ensuring that a building or a building design conforms to constraints imposed on them by governments. The Code can be viewed as a list of constraints that must be satisfied. In general, inspectors are familiar with the entire document, checking for such things as fire exits, plumbing, and structural work. Inspectors, however, will vary in their area of expertise; consequently, conformance to the Code may vary in different localities and between different inspectors within a single locality. Although the authors of the Code strive for clarity, there are ambiguities and subtleties in the document; as a result, inspectors may interpret sections of the Code differently.

Architects are primarily responsible for the design of buildings. Design includes the provision and layout of space and facilities for a specified task. For the architect, the Code provides constraints on the amount and layout of spaces, the materials from which a building can be constructed, and the type of facilities required. The Code can be viewed as a collection of mandatory requirements that must be present in a given design and minimum specifications that must be met or exceeded by the design. Alternatively, it can be viewed as a set of constraints that must not be exceeded without certain penalties. An example of a threshold type of constraint would be the floor area in certain types of occupancies. Beyond a certain threshold area sprinklers are needed for fire protection. A cost penalty is thus imposed for exceeding a certain floor area (this aspect of design is not always easy to represent).

For the engineer, whose task is primarily to implement the design of architects, the Code is a prescriptive document. The Building Code provides the engineer with limitations on the materials used for constructing buildings, methods for calculating gravity and lateral loads on buildings, as well as references to other codes and performance-based standards. Notice that the idea of imposing con-

3.1.8.1.(6) Every *firewall* shall extend from the ground continuously through all *storeys* of a *building* or *buildings* so separated, except that where a *firewall* is located above a *basement storage garage* conforming to Article **3.2.1.2.**, the *firewall* may terminate at the floor assembly immediately above the *storage garage*. (See also Sentence **3.1.8.1.(10)**)

Fig. 2. Section of the code that has been changed since the last edition.

straints on the construction of buildings is common to all four user groups. This fact is significant if computer-based building codes are to be coupled to computer-aided design (CAD) systems. The scope of this essay, however, is limited to information retrieval.¹

A study by James [3] indicates that the time for an individual to become proficient in the use of the Building Code varies from as little as 6 months to 10 years. This large variance is predominantly due to the nature of the user groups. For example, building inspectors use combinations of memory and hard-copy access. Designers tend to use aids such as annotations, photocopies of other portions (to minimize page turning), page markers, and check sheets to aid them in navigating through the document. The diversity of the user group suggests that any type of electronic representation needs to be flexible enough to be tailored at will by each specific user. One mechanism for providing these requirements is through the use of hypertext systems.

3 Hypertext

The concept of hypertext dates back several decades.² It was discussed in 1938 by H.G. Wells [5] and more fully expounded by Vannevar Bush [6] in 1945. Bush's system, Memex, was an imaginary device having the storage capabilities of a modern computer. Information was accessed by using associative links, the next item in a sequence being that which is suggested by the association of thoughts. The term *hypertext* was coined in 1974 by Nelson [7] and can be viewed as an electronic means for enhancing idea processing. The power of hypertext can be brought to bear on areas such as reading (navigation through large unstructured libraries of information), annotating (recording ideas generated dynamically while reading text), collaborating (multiple authorizing of complex documents), and learn-

¹ This is due to the fact that most current hypertext applications cannot be used for design because of the inherent limitations of the implementations. Theories of hypertext, however, have no such limitations and are closer to more familiar paradigms such as frames or objects. The verb design is defined here to be an interactive design in an integrated CAD system. The final result would either be checked automatically for Code compliance or compliance would be ensured as the design task proceeded by actively applying constraints. A small system developed by Mitusch [2] in Norway illustrates how a limited implementation such as HyperCard can be used for design purposes.

² An excellent hypertext bibliography has been compiled by Nielson [4].

Fig. 3. (a) One-dimensional (a novel), (b) two-dimensional (a manual), and (c) three-dimensional text (hypertext).

ing (personalized structuring of bodies of information) [8].

Flat text (or paper text) provides one or two dimensions of information processing. Text can be structured linearly or hierarchically. Hypertext can be viewed as a three-dimensional web of nodes (modules) joined together with links. Various forms of text are shown in Fig. 3.

4 Nodes

Nodes are intended to be information-rich chunks that at their finest level of granularity represent single concepts or ideas. In general, nodes are structureless blanks or slates where an author could place text (or graphics). The text could be fragments or a whole document. Nodes can be typed, where typed nodes facilitate the distinguishing of nodes at a glance. Color, size, or icons may be used to type nodes. The concept of typing is particularly useful when one considers adding internal structure to the node. Semistructured nodes are templates that constrain the author allowing only appropriate slots or labeled fields to be filled. By aggregating

several nodes into a composite, a considerable amount of related information can be collapsed into a single structure.

5 Links

A hypertext link is like an electronic footnote, an endnote, or a parenthetical phrase.³ Links can be either predefined or user-defined. Predefined or structural links constitute a mapping between a conventional document and a hyperdocument. Hypertext links can perform several functions. They can connect document to a reference document, connect comments and annotations to a document, provide organizational information, connect successive pieces of text, and connect tabular and graphical entities to other nodes. Conklin [9] suggests three general classes of links: referential (a nonhierarchical link), organizational (a hierarchical link), and keyword. A user-defined link has an arbitrary type and connects whatever nodes the user wishes to connect.

6 Reading a Hyperdocument

Reading a piece of hypertext (a hyperdocument) can be more complicated than reading normal text. Reading can be viewed as goal-oriented navigation through a multidimensional space. The reader is provided with a set of tools to navigate through this space. Generally, these tools fall into three categories: links, search and query languages, and graphical browsers. Navigation through a hyperdocument is achieved primarily through the use of links. The reader follows these links from node to node until the desired goal is attained.

Search and query languages (SQLs) provide the reader with a powerful tool for reorganizing and focusing the information in a hyperdocument. These languages can be used to select nodes, returning a list of nodes matching a query. The concept of filters, whereby specific information in the nodes could be suppressed or highlighted, can be implemented using SQLs. A promising approach in the theory of hypertext is the ability to create virtual nodes using an SQL. This is directly analogous

to the creation of virtual views in a relational data base.

Another powerful method of navigation involves the use of graphical browsers. An interactive browser allows the reader to view a portion of the document in relation to the overall context, as well as all the links to and from a node of interest. Nodes can be expanded or compressed and the network zoomed in and out as required. Although graphical browsers do not solve all the problems related to navigating through a hyperdocument, they do provide the reader with a picture of the overall context, reducing the likelihood of getting "lost."

7 Hypermedia

Hypermedia is an extension of hypertext [10]. The nodes in a hypermedia system would include photographs, graphics, films, and sounds (olfactory, gustatory, and tactile stimuli would naturally follow). The links in such systems could also pass control to other application software, such as spread sheets, data base applications, and communications programs. For the remainder of this essay the term *hypertext* will actually refer to hypermedia. An example of "pseudo-hypermedia" is shown in Fig. 4a. The figure shows a stylized representation of a imaginary path through the Building Code and is not modeled after any particular implementation of hypertext. Similarly, Fig. 4b shows a sample of hypertext from the Symbolics' Document Examiner.

The two figures also demonstrate two different approaches of window control strategies, namely, user-controlled overlapping windows and a window-tiling system. An initial field trial of a hypertext-based Building Code indicated that some users found the overlapping window confusing [11]. Some studies have shown that users prefer the nonoverlapping tiled approach; however, this type of research is in its infancy and the results are only preliminary [12].

The successful implementation of a hypertext-based Building Code hinges on the development of

Fig. 4b. A sample of hypertext from the document examiner. The system combines links, a search and query language, and graphic browsers. In this case the reader has requested information on the Document Examiner system. A list of all candidate documents was displayed in the Current Candidates window. Then the reader selected the desired documents, which were placed in the Bookmarks window, and the current document was marked with an arrow. The complete document, including graphics, was then shown in the viewer window. User input was achieved using a combination of mouse and keyboard devices.

³ From an implementation point of view links are merely pointers to other items; however, when viewed from a higher level of abstraction links take on degrees of functionality, such as performing a task when a link is traversed (the rhetorics of arrival and departure).

Fig. 4a. A mock-up of a hypertext building code, showing an imaginary trace through the Building Code. The reader has inquired about firewalls. The sequence of user inquiries is shown by the overlapping windows. The trace begins with a list of topics concerning firewalls. An inquiry on penetration through service openings led to a split in the information path, specifically information dealing with combustible materials or projections. The reader selected the projections options, and all the information regarding projections was displayed [Sentence 3.1.8.1.(15)]. The graphic icon in the lower right-hand corner of the window indicates an available graphic. A definition of the term *firewall* was displayed when the reader selected the word *firewall*.

Figure 1. Upon selection, the Document Examiner displays all the books in the document set.

window control strategies acceptable to all user groups. Allowing the reader to select an appropriate window control strategy appropriate to a particular task may solve the problem of using one global strategy for diverse tasks.

8 Issues

Several issues regarding hypertext remain to be resolved. Disorientation can occur when the reader (1) is lost in the network, (2) has forgotten the original goal of search, (3) cannot locate a piece of information, or (4) cannot access a piece of information. Browsers do not entirely solve the disorientation problem. Data base search and query techniques may help; however, the results of query may be a long list of nodes of indeterminate relevance, or an empty list.

Cognitive overhead [9] is the additional effort and concentration necessary to maintain several tasks at the same time. Multiwindow applications increase the cognitive load on the user; although more time is spent manipulating the interface, the overall productivity is increased in multiwindow applications [12].

In addition, cognitive overhead can manifest itself during the authoring of hyperdocuments. An excellent example of this is documented by Slatin [13] as he attempts to insert a commentary in a hypertext version of a poem. Much consideration is given to the structure of the document (the creation of nodes and their appropriate links, potential means of access, whether or not the structure conforms with accepted rules of composition and referrals, and the context in which the annotations will appear). This activity is occurring when the author should be concerned only with the content of the document and not with the structure (i.e., the author should be aware of the structure but not actively developing it). There also exists the difficulty of representing themes or highly interrelated ideas in nodal format. It is often difficult to construct rhetorically neutral free-standing prose. There is no predetermined answer to the question of what constitutes a node. In fact, the definitions of nodes are inextricably tied to the links that join them [13].

The features of an idealized hypertext system are outlined as follows [9]:

1. A data base consisting of a network of textual (or graphical) nodes comprising a hyperdocument
2. Windows corresponding to nodes on a one-to-one basis. The windows should support standard window operations such as scrolling
3. Navigation through the hyperdocument using links, search and query methods, and graphical browsers;
4. Ability to create new nodes and links
5. An arbitrary number of link icons for a given node

9 Why Hypertext Is an Appropriate Medium for Code

Hypertext's characteristics make it an ideal medium for representing the Building Code. The Code is not a traditional book meant to be read sequentially from cover to cover. It is a reference document, somewhat like a user's manual, and is intended to be accessed in a nonsequential, goal-oriented fashion. In addition, it is structured hierarchically and is composed of items that represent discrete concepts or ideas. Basic units of the Code are articles, sentences, clauses, and subclauses that represent relatively well-defined concepts within the context of more general building blocks of the Code. Concepts and ideas become more generalized as the reader moves up the hierarchy to subsections, sections, and parts. The links within the document are mainly explicit, unlike traditional prose that has many implicit links (poetry, e.g.). These characteristics lend themselves well to a three-dimensional hypertext representations.

There are several different groups of users of the Code. Within those groups individuals may or may not share the same view of the document. Individuals working with the document may have different goals in accessing it. An inspector reviewing plans for a house would have different goals and strategies from those of an inspector reviewing plans for high-rise buildings. The reader can personalize the Building Code in several ways. By following an arbitrary path through the document, or filtering information in the nodes (returning all nodes relevant to buildings with occupancy type B, for example), the reader can generate a personalized view. The creation of virtual nodes by using search and query languages allows the user to create virtual documents based on arbitrary specifications.

The tailoring of the document to specific end uses or goals does not rest solely in the ability to associate concepts or ideas freely via hypertext links. Users of the Code regularly annotate, place bookmarks, photocopy relevant portions for insertion elsewhere, create checklists, and otherwise tailor the document to their own needs. Hypertext systems allow the reader to perform all these tasks

effectively. Annotations can be added by creating new nodes containing the annotations and linking them to specific points in the document. Checklists can be generated by tracing a path followed through the document. Bookmarks and reminders can be placed as iconic links on various nodes, and references can be accessed repetitively without redundancy in data storage.

A hypertext version of the document would give a user access to all the reference documents specified in the Code as well as supplemental information, such as climatic data. The opportunity also exists to attach information such as rulings and interpretations, bulletins, and other ancillary information. Wherever appropriate, graphical or audio-visual information could also be included in the network [14].

This may seem to be a rather grand vision; however, the technology exists to construct a hypertext version of the Building Code and to deliver the document on affordable workstations.

10 Compatibility and Version Control

The vision of the HyperCode is as follows. The general framework for the HyperCode is distributed to the users. Modifications in the form of revisions are issued on a yearly basis, much as patches are released for commercial software products. Every 5 years a new version is released, which the user must purchase. Changes from the previous version of the Code would be marked with a change bar, as is currently done in the paper edition. Revisions would not include any structural changes to the document, only the modification of individual nodes. Thus, the mechanism for handling revisions can be relatively straightforward. The revisions would simply replace the existing nodes. The existing nodes would remain in the system and would be linked to the body of Code through the change bar links. A revision to the current edition Code is shown in Fig. 5.

Fig. 5. An example of a revision. The revised nodes are shaded. Older nodes are linked to the revisions (shaded nodes) via change links.

Different editions of the Building Code can entail major semantic and structural changes. If different versions of the Code are to be incorporated into one document, then the issues of compatibility and updating must be dealt with. When a change icon is traversed, bringing the reader into the previous edition, the text of the changed node and the reason for the change would be displayed. If the node in the older document was linked to other nodes, the reader might be tempted to follow the links. If the structure of the document has changed, then the links might lead nowhere, or worse, to incorrect locations, possibly leading to disorientation.

There are two possible solutions to this problem: (1) maintain the previous n editions of the Building Code, the links between each independent code being established by the change icons, and (2) support only the current edition, and provide read-only text for the past editions (i.e., do not support the links). Figure 6 illustrates these two ways of handling document updates.

The first alternative allows the reader to access previous Building Codes if they need to be consulted (structures built to conform to previous editions, for example). Care must be taken to ensure that the reader will not be confused between versions, especially when a complex path crossing over many editions is followed. The second alternative, although easier to implement, does not allow the reader to access fully previous Building Codes for historical purposes. It also does not totally eliminate the version control problem. If a reader has annotated the document, it is in the reader's interest to keep those hard-won pearls of wisdom intact. Indeed, James [3] points out that in certain design offices checklists have been handed down through the years. How then does a user transfer this knowledge from one version of the document to the newest version of the document? Transferring this type of information manually could involve considerable effort, or it may simply involve the relabeling of referential links. What is required is a method of labeling the semantic content of a code item and a

Fig. 6. Making structural changes in a document. (a) Example of updating while maintaining separate versions. The revised nodes are shaded and represent changes from the previous edition. The links between each edition are established via the change bars (arrows). (b) Example of updating while maintaining only the current version. The revised nodes are shaded and represent changes from the previous edition. The links between the new nodes and changes nodes are established via the change bars (arrows). The old nodes are read-only.

user-defined node and then relinking the user node with a code item of the same or similar meaning. Work on developing hypermedia systems that address the issue of version control is ongoing. A few large systems have implemented some sort of version control capability, keeping track of the history of nodes as well as considering changes in the network as a whole [15, 16].

11 HyperCode Possibilities: The Next Generation

How could a HyperCode be extended or expanded? Hybrid systems, combining elements of artificial intelligence and hypertext concepts, could extend the power and flexibility of a hyperdocument. Hypertext systems could also prove to be powerful tools in examining the structure and content of the Building Code as well as a useful tool in conducting user studies.

12 Hypertext and Artificial Intelligence

An analogy between hypertext and semantic networks is drawn by Conklin [9], in which hypertext

nodes can be viewed as representing concepts or ideas, and the links can be viewed as the semantic relationships between the nodes. The concept of semistructured nodes is highly reminiscent of the concept of frames [17]. If the nodes in a hyperdocument were viewed as frames, then powerful concepts such as procedural attachments and daemons could be added to the system. Similarly, if one views the nodes as objects in an object-oriented system, then all the power and flexibility of the object-oriented programming paradigm becomes available. Halasz [18] indicates that at a high level of abstraction hypertext, frame, and object systems represent nearly identical data models, each based on the notion of typed, slotted entities that form a network structure.⁴

An object-oriented hypertext system would allow for multiple access paradigms. Readers could consult the HyperCode for informational retrieval purposes; however, more sophisticated reasoning could be achieved by accessing the class hierarchy of the object data base and the methods attached to

⁴ In fact Notecards, a well-known large hypertext system was implemented in a LISP object-oriented programming system (LOOPS).

the objects. A reference model of a building could be created from information contained within the Code. The model would comprise components and classifications and would be represented as objects. Technical requirements of the Building Code could also be represented as objects. These objects would be classified as constraints. Incorporating the knowledge contained within the Code into CAD systems would allow designs to be generated within the constraints of the Code or to be verified automatically after the fact. The building of hypertext systems using the object-oriented paradigm parallels the ongoing work to develop object-oriented intelligent CAD systems.

13 Codes-Related Research

The ability of hypertext systems to store the paths followed by a reader would enable designers of code to study how the document was used. Problem areas could easily be identified and corrected by examining the traced paths. User-defined shortcuts and structures could also be incorporated into the Code. The frequency with which nodes in the document are accessed, the length of a path to a node (the average path length for the entire document), the size of node, the number of cross-references, and any number of data could be recorded and examined with the goal of improving the efficiency of information retrieval.

A slightly more esoteric approach would be to use hypertext systems to explore ambiguities and differing interpretations of the Code. Systems could be designed around the concept of issue-based information systems (IBIS) [9–19]. Issues could represent problem areas in the Building Code. Persons would have positions on various issues, and those positions would have to be supported by arguments. Issues, positions, and arguments would be posted to the hyperdocument. Other individuals could respond, in a multiuser environment, to various arguments possibly posting counterarguments. Some consensus or resolution of the problem from the computer-based discourse could be posted as a revision to the Building Code.

14 Conclusions

The National Building Code of Canada is a complex document. It consists of a large number of prescriptive rules, linked to other rules within the document and refers to a large number of other documents.

The document itself refers to, or contains, voluminous tables, and some graphical information. Many users annotate copies of the document and would prefer much more graphical information. Users also alter the sequence of items in the document or produce summary sheets of items to be checked. The Building Code itself is a form of hypertext with the reader performing the tasks of a central processing unit.

Within the context of information retrieval, hypertext is an ideal mechanism for representing the Building Code. The ability to navigate through the document, following whatever links are appropriate to the task, duplicates what is currently done manually by humans (associating ideas). A record of those trails or associations is kept. Hypertext systems allow the reader to retain the context in which information accessing is occurring, keeping relevant information on the screen at all times, or instantly accessing information viewed before.

The concept of a personalized view of a document allows a variety of users of the Code to tailor the document to their goals. Tracing the path navigated through the document, annotating the document with notes and graphics, or generating new documents using search and query methods allow each user to construct his or her own specific version of the Code.

A hypertext version of the Building Code would serve as starting point for an expanded document. The inclusion of artificial intelligence concepts into hypertext systems would enable the creation of design systems built on object-oriented paradigms.

References

1. National Building Code of Canada (1985) Issued by the Associated Committee on the National Building Code, National Research Council of Canada, Ottawa, Ontario
2. Mitusch, P.D. (1988) Expert system for the Norwegian building regulations—A new approach. Norwegian Building Research Institute, P.O. Box H 123 Blindern, N-0314 Oslo, Norway
3. James, K. (1989) User Study of the National Building Code, Contract Report, Contract No. 988-44222D, National Research Council of Canada, Institute for Research in Construction, Ottawa Ontario
4. Nielson, J. (1989) Hypertext bibliography. *Hypermedia* 1:1, 74–91
5. Wells, H.G. (1938) *World Brain*. Garden City, New York: Doubleday, Doran & Co.
6. Vannevar, B. (1945) As we may think. *Atlantic Monthly* 176:1, 101–108
7. Nelson, T.H. (1974) *Dream Machine: New Freedoms Through Computer Screens—A Minority Report*. Issued with, *Computer Lib: You Can and Must Understand Computers Now*. Chicago, IL: Hugo's Book Service

8. Carlson, P.A. (1988) Hypertext: A way of incorporating user feedback. In: Text, Context and Hypertext (Ed. E. Barrett), pp. 93-110. Cambridge, MA: MIT Press
9. Conklin, J (1987) Hypertext: An introduction and survey. *Computer* 20, 17-41
10. Young, J.S. (1986) Hypermedia. *MacWorld* March, 116-121
11. Vanier, D.J. (1989) Computerization of Building Regulations. In: Proceedings of the International Conference on Municipal Code Administration Building Safety and the Computer. Manitoba Building Officials Associations Inc., Winnipeg Manitoba, pp. 43-62
12. Lifshitz, K.; Schneiderman, B. (1987) Window Control Strategies for On-Line Text Traversal. Department of Computer Science and Human Computer Interaction Laboratory, University of Maryland, College Park, MD (correspondence with B. Schneiderman)
13. Slatin, J.M. (1988) Hypertext and the teaching of writing, text, context and hypertext (Ed. E. Barrett) pp. 111-129. Cambridge, MA: MIT Press.
14. Younggren, G. (1988) Using an object oriented programming language to create audience-driven hypermedia environments. In: Text, Context and Hypertext (Ed. E. Barrett), pp. 77-92. Cambridge, MA: MIT Press
15. Delisle, N.; Schwartz, M. (1986) Contexts—A partitioning concept for hypertext. In: Proceedings of the Conference on Computer Supported Cooperative Work, Austin, Texas, December 3-5, pp. 147-153
16. Goldstein, I.; Bobrow, D. (1984) A layered approach to software design. In: Interactive Programming Environments (Eds. D. Barstow; H. Schrobe; E. Sandewall), pp. 387-413. New York: McGraw-Hill
17. Minsky, M. (1975) A framework for representing knowledge. In: The Psychology of Computer Vision (Ed. P.H. Winston). New York: McGraw-Hill, pp. 211-277
18. Halasz, F.G. (1988) Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM* 31(7), 836-852
19. Rittel, H.; Webber, M. (1973) Dilemmas in a general theory of planning. *Policy Sci.* 4