ELECTRONIC PUBLISHING, VOL. 3(3), 125–140 (AUGUST 1990)

Hypertext writing and document reuse: the role of a semantic net

ROY RADA

Department of Computer Science
University of Liverpool
Liverpool L69 3BX, UK

e-mail rada@liverpool.ac.uk

SUMMARY

When document components are classified and then recombined during document reuse, a semantic net may serve as the classification language. A theory of analogical inheritance, applied to this semantic net, guides the reorganization of document components. Authors index paragraphs from various sources with node-link-node triples from a semantic net and then use programs to traverse the semantic net and generate various outlines. The program examines node and link names in deciding which path to take. This paper describes how these techniques helped the author to reuse parts of an existing book to write a new one.

KEY WORDS Semantic net Document reuse Linearization

1 INTRODUCTION

One of the most salient differences between text and hypertext is the abstraction of the text as a network. This paper explores the utility of an approach in which the network is viewed as a semantic net and in which traversals of the network produce new documents. In particular, this approach is applied to the problem of document reuse.

Computer writing tools should be tuned to certain users and tasks [1]. The basic strategy of reusing information to create new information is well accepted [2] and has been applied to writing [3]. In order to successfully reuse document material, some kind of classification of existing material is vital [4]. Secondary information services typically index a document into terms of an indexing language which can be considered a semantic net [5]. One hypothesis of this paper is that a semantic net can be a useful representation in document reuse.

Secondary information services answer queries by returning a set of document citations. For document reuse something more particular is wanted. Namely, the retrieved material must be organized into a sequential, cohesive document. The second hypothesis of this paper is that patterns of regularity in the semantic net can be exploited so as to generate meaningful, linear documents.

2 REPRESENTATION

The notion of a network is fundamental to hypertext. Since the first half of the

0894–3982/90/030125–16$08.00

Received 25 May 1990
Revised 2 October 1990

century, people have dreamed of hypertext systems [6], but the best representation for hypertext remains unclear. Some researchers focus on a logic-based approach to hypertext, some build a hypertext model on top of Petri net formalisms [7], but the basic structure remains that of nodes and links with attributes [8].

2.1 Semantic net infrastructure

A semantic network is a graph where natural language terms have been used to label the nodes and links. In a semantic network concepts are represented by terms and their relationships to other terms in the network. For example, to place the concept 'hypertext' in a semantic net, one might begin by saying that it contains documents, runs on computers, and serves users. The link types are 'contains', 'runs on', and 'serves'; the nodes are 'hypertext', 'documents', 'computers', and 'users' (see Figure 1). A semantic net lends itself to graphic display, and its meaning tends to be intuitively, if not formally, clear.

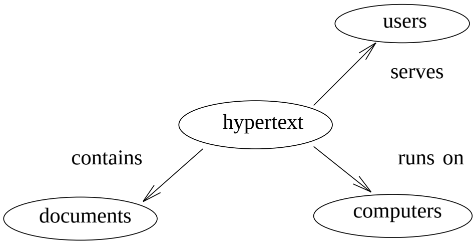
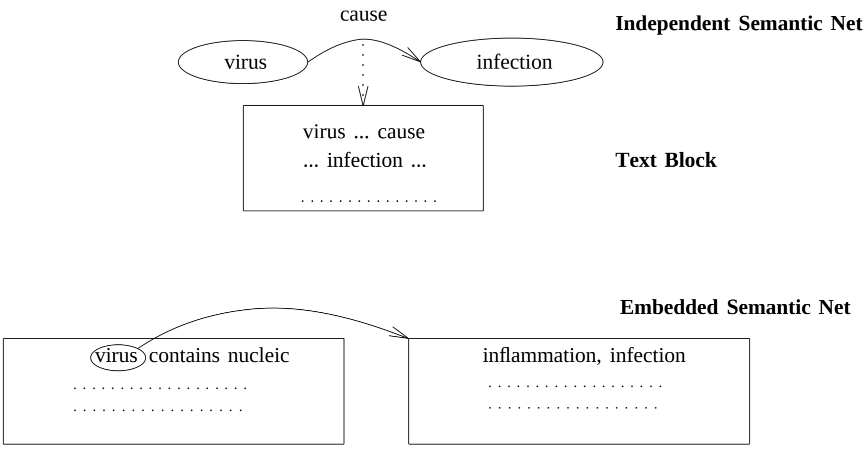


Figure 1. Semantic net example



Above: Sketch of independent semantic net a level above the document itself. A Text Block is associated with the link which connects the two nodes.

Below: Sketch of semantic net embedded within the document. A Text Block is associated with the node 'virus' which is also a word embedded within another Text Block.

Figure 2. Independent versus embedded semantic net

Some semantic links manifest inheritance. For instance, if the network connects the node 'student' to the node 'person' with the link 'is a', then one can infer that the properties of 'student' are inherited from those of 'person'. Inheritance conveys transitivity. If a person is an animal, then by transitivity a student is an animal.

The semantic net of hypertext may be *independent* or *embedded*. In the 'independent' case the nodes and links are tagged with terms [9]. The nodes or links point to text blocks, but the semantic net can be seen without necessarily seeing the text blocks. In the 'embedded' case a text block is at the end of a link (see Figure 2). In traversing an embedded semantic net hypertext, the user has to visit a text block. This analysis can be significantly extended as any quantity of information may be identified as a node [10].

2.2 Translation into a database management system

The independent semantic net representation was implemented in a relational database system, as a book was imported into the database. Two classes of students at George Washington University over two semesters worked with the database and augmented the semantic net on-line. A computer program was developed to read a book (called *Machine Learning: Expert Systems and Information Retrieval*) [11], exploit the markup language (the Unix Document Workbench), and translate the book into a relational database management system (SQL/DS). One relation in the database was for paragraphs, and another was for a semantic net that included the book's outline.

The semantic net supported hierarchical and non-hierarchical relations between terms. Each relation had two fields, as in Hierarchy(term$_1$, term$_2$). Another relation, the Point relation, associated a paragraph with a term and took the form Point (term, unique identifier). Through a combined view of the Hierarchy and Point relations, a user could follow a network term to the text about that term. An interface program helped students add paragraphs to the database and manipulate the semantic net.

The students had difficulty augmenting the semantic net. In particular the meaning of a term or link in the semantic net was often unclear to them. To provide further guidance a Definition Table was created. For each term, students were asked to provide a definition, an example of its use, or explanations of the term's relations to other terms. For example, the term 'expert system' was described in the Definition Table with hierarchically and non-hierarchically related terms (see Figure 3).

Definition table	
Term	**Textual description**
expert system	Hierarchically more general term is artificial intelligence (see paragraph 127 for text that discusses artificial intelligence and expert systems). Non-hierarchically related term is machine learning (see paragraph 456 about acquiring rules through interaction with user).

Figure 3. The Definition Table in the hypertext database contained a free-text field similar to this one (the authentic version is not worded this carefully)

The ‘quality’ of the links which had been created in the semantic net were compared for the situation in which no Definition Table existed and for the situation in which a Definition Table did exist. This comparison was made by two scientists studying the node-link-node triples and assigning a quality score to each triple [12]. The use of the Definition Table led to better node-link-node triples.

2.3 Paragraphs on edges

In the next phase of the investigation, paragraphs were extracted from many other documents and indexed by node-link-node triples. In this way the indexed paragraph provides an example or definition of what the node-link-node triple means. A node-link-node triple is a richer index for a paragraph than just a node. By placing paragraphs on the links, one can say that a paragraph is about a certain relation between two nodes – not just about a node. Arbitrary link or relation names were allowed – not just hierarchical and non-hierarchical relations.

The semantic net model may be seen as a set of link objects. Each link object specifies some source node, link type, target node, pointers to paragraphs, and perhaps other attributes. A group of link objects with the same source node form a frame (see Figure 4) [13].

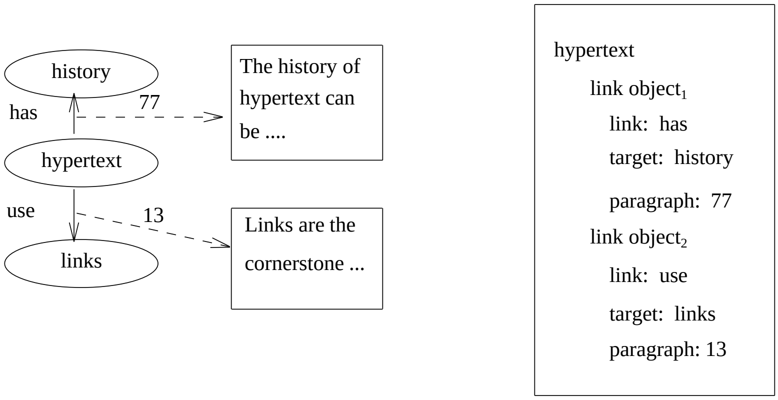


Figure 4. Graphical and frame representation are given side-by-side to show the frame representation

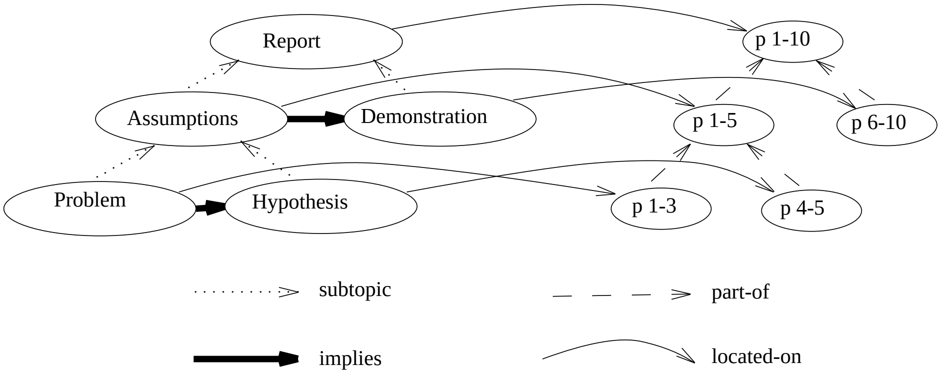
2.4 Models of the domain

In the top-down approach to indexing, the semantic net is first elaborated and then the paragraphs are indexed. In the bottom-up approach the paragraphs are first collected, and the semantic net is built as the paragraphs are indexed. The better approach is a combination of the top-down and the bottom-up approach [14]. To build and maintain a semantic net, indexing of paragraphs and semantic net construction go hand-in-hand. If a paragraph is to be indexed but no component of the semantic net relates to that paragraph, then the semantic net is augmented [15].

The purpose of the semantic net is to give people an overview of or handle on the content of the text. To this end, the semantic net must itself convey a meaningful model of the world or must somehow present patterns to the user that are easily understood. The user must be able to predict from one part of the semantic net what is likely to be in another, analogous part of the semantic net.

In studies of the indexing languages of secondary information services, patterns of analogical inheritance have been observed [16]. Analogical inheritance extends the notion of inheritance by saying that the attributes of a node should be related to the attributes of its parent in some systematic way. If the attributes were the same, then simple inheritance would attain.

A semantic net may be formally represented as a directed, labeled graph with nodes $V = \{v_{0,}\ v_{1,}\ \ldots, v_n\}$ and links $\{(v_i, v_j, l_k) \mid v_i, v_j \in V ; l_k \in labels\}$ where $labels = \{ is-a, part-of, \ldots, cause \}$. The predicate $P(vi, v_j, f)$ is true, if and only if v_i and v_j are connected by a path f of labels. Analogical inheritance on paths f, g, and h is present to the extent that $(\forall v_i, v_j, v_{i'}, v_{j'} \in V)\ (P(v_i, v_j, f) \wedge P(v_i, v_{i'}, h) \wedge P(v_j, v_{j'}, h)) \rightarrow (P(v_{i'}, v_{j'}, g))$ is true [17]. If two nodes are related by f and are source nodes for the path of labels h, then the target nodes for the path h should be related by g. Three examples of fuzzy analogical inheritance can be seen in the outline of Figure 5 [18].



Manifested here are three examples of analogical inheritance along the f, g, and h functions:

* subtopic, part-of, and located-on,
* subtopic, $(subtopic^2)^{-1}$, and implies, and
* subtopic, $(subtopic^2)^{-1}$, and $implies^{-1}$.

For instance, for ‘subtopic’, ‘part-of’, and ‘located-on’ functions, the following is true: P (Assumptions, Report, subtopic) ∧ P (Assumptions, p 1-5, located-on) ∧ P (Report, p 1-10, located-on) → P (p 1-5, p 1-10, part-of).

Figure 5. “Outline”: example of an outline viewed as a semantic net

Analogical inheritance can be used to guide the augmentation of a semantic net. For instance, if the existing semantic net notes that ‘hypertext includes text’, ‘hypertext has principles’, and ‘principles include database principles’, then an analogical inheritance guide might note that the addition of ‘text has database principles’ would increase the amount of analogical inheritance or regularity in the network.

3 EXPERIENCES WITH SEMANTIC NET CONTENT

The paragraphs being added to the database became part of a new book on hypertext entitled *Hypertext: from Text to Expertext* [19]. When either looking for blocks of text from other documents to include in the new document database or when considering new additions to the semantic net itself, preference was given to additions that would make the semantic net manifest more analogical inheritance. One of the difficulties in applying the structuring principle was the diversity of disconnected link and node types present in the semantic net. To this end efforts were made to control this diversity.

3.1 From *Roget's Thesaurus*

One source of structuring information is *Roget's Thesaurus* [20]. *Roget's* contains over 1000 word types, and for each type different forms of speech and numerous synonyms are offered. Furthermore, the thesaurus embeds each word type in a hierarchy.

Imagine that one wants to say that 'hypertext includes text' but is uncertain about the wording. One can go to *Roget's* index and under the word 'include' find an alphabetically sorted list of words that are alternatives to 'include'. The words listed under 'include' in Figure 6 are distributed in *Roget's* hierarchy as illustrated in Figure 7. With this hierarchy one can determine commonalities among terms. If, for instance, one wanted a word related to '2.II.C.236', one could move upward in the classification to '2.II.C.', and then look at the descendants of that category; or go still higher to '2.II.'.

Sample from Index

- include
  - combine 52.3
  - comprise 76.3
  - enclose 236.5
  - internalize 225.6
  - join 47.5

Figure 6. The index at the back of Roget's lists over 1000 words, each of which has indented underneath it another list of words. The number after a word refers to its location in the hierarchy of words which Roget's contains

- 1. Abstract Relations
  - 1.III. Quantity
    - 1.III.C. Conjunctive Quantity
      - 1.III.C.47 Joining (*join* 47.5)
      - 1.III.C.52 Combination (*combine* 52.3)
    - 1.IV. Order
      - 1.IV.D. Distributive Order
        - 1.IV.D.76. Inclusion (*comprise* 76.3)
- 2. Space
  - 2.II. Dimensions
    - 2.II.C. External and Internal Dimensions
      - 2.II.C.225. Interiority (*internalize* 225.6)
      - 2.II.C.236. Enclosure (*enclose* 236.5)

Figure 7. Roget's hierarchy is meant to cover the universe of word types. A small extract from the hierarchy is shown here so that one can see the path from the root of the hierarchy to each of the words in Figure 6

In one phase of semantic net development, every new link label was also created as a node label. These node labels were further connected in a hierarchy that duplicated that in *Roget's*. In this way the author can create any link labels but still look for patterns of analogical inheritance by mapping the link into similar links according to the hierarchy in *Roget's*.

While several person-months were invested in the development of a semantic net whose links were labeled with terms from *Roget's*, the profit from this investment was not substantial. This poor performance may be due to a paucity of tools. For one, the researcher did not have *Roget's* on-line and needed to manually enter into the computer the relevant relations from *Roget's*. For another, the computer programs which analyze semantic nets for patterns of inheritance were not usable, while this version of the semantic net was being developed. Detecting patterns in a large, complex semantic net is difficult to do without the aid of computer programs.

3.2 Patterns in names

In other exercises, the goal was to reduce the variety of link types so that visual inspection of the semantic net would more readily lead to recognition of repeating patterns. In one version of the semantic net the most common links were 'include', 'use', and 'has', in that order (see Figure 8). These link types do not correspond to those developed either for other subject areas, like toxicology, or for other activities, such as discussion. In the area of toxicology, a hypertext has been prepared, and each toxin is described along with the following attributes: common name, molecular weight, molecular formula, characteristics, source, and toxicity [21]. For 'discussion' systems, certain node and link types may be identified [22]; for instance, an 'Argument' node may have a 'respond' link to an 'Issue' node. The work with the *Hypertext* book did not necessarily identify a set of link types that were either generic to producing a book or to the topic of hypertext. The set of relevant link types will depend on the model of the domain to be conveyed.

Distribution of links	
link name	number of occurrences
include	227
use	158
has	147
example	67
apply	50
need	48
extend	26
imply	25
past	18
advantage	13
disadvantage	11
is	4

Figure 8. Only 12 types of links were used in the semantic net that supported the final draft of the Hypertext book. The complete listing of linknames and their frequency is shown here

The frequency distribution of source and target node names served as a guide to augmentation of the paragraph database and of the semantic net. As a rule of thumb no source node name should occur in just one link object nor in many link objects. This follows the traditional wisdom that a menu of about 4 to 13 items is most manageable by people [23]. A survey of node names revealed about 200 unique source node names (see Figure 9). The average frequency of occurrence was about 3, and the most frequently repeated name was ‘microtext exercises’ which occurred 12 times. (‘Microtext’ is a neologism which means ‘small-volume hypertext’.) An analysis of the target node names revealed about 600 unique target node names with most occurring just once (see Figure 10). The average frequency of occurrence of a source node is greater than that of a target node, and a source node always has more than one target.

The names of nodes revealed an interesting pattern. Several ‘source node names’ were prefixed to other ‘source node names’. For instance, the semantic net contained a source node called ‘Microtext’ and another called ‘Macrotext’. (‘Macrotext’ is a neologism for ‘large-volume hypertext.) For each corresponding frame, most of the ‘target node names’ were prefixed by the ‘source node name’ (see Figure 11). Such patterns, once detected, suggest guidelines for other parts of the semantic net. For instance, sibling nodes (nodes which have some hierarchical relation to the same node) might be expected to have the same pattern of ‘target node names’. By example, source nodes that are siblings to Microtext and Macrotext might be expected to have target nodes with

Sample from source node distribution	
source node name	number of occurrences
Microtext Exercises	12
Microtext Requirements	11
Macrotext Exercises	10
...	..
Augmentation System Results	2
Electronic Yellow Pages	2

Figure 9. An excerpt from the list of source node names which shows some of the most and least frequent names

Sample from target node distribution	
target node name	number of occurrences
framework	18
motivation	13
history	11
...	..
1960s	1
1970s	1
1980s	1

Figure 10. An excerpt from the list of target node names which shows some of the most and least frequent names

suffixes including ‘history’, ‘principles’, and ‘systems’, and exactly this happened. The nodes ‘grouptext’ and ‘expertext’ were siblings to ‘microtext’ and ‘macrotext’ and had frame representations that repeated the patterns of ‘history’, ‘principles’, and ‘systems’.

- Microtext
  - Microtext History
  - Microtext Principles
  - Microtext Systems
  - Text to Microtext
- Macrotext
  - Macrotext History
  - Macrotext Principles
  - Macrotext Systems
  - Hypertext to Hypertext

Figure 11. Source and target node names for frames for Microtext and Macrotext

4 LINEARIZATION

The success of the document reuse was measured by its support for the production of a new book entitled *Hypertext*. While the new book can be viewed as a hypertext because it has a semantic net to which paragraphs are attached, print on paper remains the dominant medium for the delivery of books, and *Hypertext* has also been generated in paper, linear form. The value of printing a linear document from a database has been appreciated for a long time. The FRESS system of the 1970s supported printing from a document database [24]. The commercial hypertext systems KMS, Guide, and Hyperties all support print features [19].

This paper presents a method for automatically generating a document by a traversal of the semantic net. The method has been automated because

- the semantic net is so large that to traverse it by hand would be laborious, and
- in the process of revision one wants to see the semantic net and see the linear form repeatedly.

A program allows one to frequently modify the hypertext database and test its consequences for the linear form.

4.1 Depth-first traversal

The obvious methods of traversing a graph are breadth-first and depth-first. A breadth-first traversal goes from one node to all the nodes directly connected to it and then resumes the process at one of those connected nodes. A depth-first traversal goes from a node to one of the nodes directly connected to it and then immediately continues by visiting a node that is connected to the last visited node (see Figure 12). In these traversals, which target node to visit next of those in one frame and not yet visited is determined randomly. Since this random approach may be inadequate, the Link Objects have an attribute called ‘Order’. With the numeric value assigned to this ‘order’ attribute, the

hypertext
includes text p1
includes microtext p2
includes macrotext p3
text
includes history p4
includes principles p5
includes systems p6

A paragraph is associated with each node-link-node; for instance, paragraph p4 is about ‘text includes history’. The breadth-first traversal from ‘hypertext’ visits the nodes text, microtext, macrotext, history, principles, and systems, in that order. The depth-first traversal goes from ‘hypertext’ to text, history, principles, systems, microtext, and macrotext, in that order.

Figure 12. Portion of semantic net with attached paragraph pointers

traversal algorithm decides which target to visit next. The traversal algorithm also allows users to specify link-type priorities. The traversal algorithm will visit the link objects in a frame in the order of their priorities. No node is visited twice.

The first program for traversing the semantic net was a combined breadth-first, depth-first one. This combined traversal first collected all the paragraphs from links emanating from a node (this is the breadth-first part) and then chose the next node by a depth-first principle. From each selected node, the process repeated itself; namely, all paragraphs on links from that node were printed, and then the next ‘depth-first’ node was visited. For instance, while traversing the semantic net in Figure 12, the breadth-depth traversal prints paragraphs in the order p1, p2, p3, p4, p5, p6.

This combined depth–breadth approach has the advantage of giving the reader an overview of what is to come, but the disadvantage of an uncomfortable jump. This jump occurs after the paragraph corresponding to the last link from a node and before the first paragraph associated with the next node. In the preceding example, by seeing p1, p2, and p3 the reader gets a good overview of hypertext. However, when p4 is reached the reader has just finished reading about some other aspect of hypertext than text. The paragraph p1 would be the natural one to be read before reading p4. To address this difficulty the algorithm was changed to a depth-first one.

The depth-first traversal has, of course, its own shortcomings. In the depth-first traversal for the above example the paragraphs would appear in the order p1, p4, p5, p6, p2, p3. This ordering is fine for the progression from p1 to p4 to p5 to p6. But the jump from p6 to p2 is large and difficult. To compensate for this the author might include remarks in p6 which somehow introduce p2. Furthermore, the hierarchical structuring of the book is typeset so as to emphasize to the reader that p2 is at the same level as p1 and not a continuation of p6.

When the semantic net is built, connections may be made which have a secondary significance. For instance, the semantic net includes a link from ‘text’ to ‘ancient history’. This link is associated with a paragraph about the ancient origins of text. In a reasonable sequence the paragraph about the history of text in the Middle Ages would come next. If the graph traversal were to follow the target node ‘ancient history’ to a source node ‘ancient history’ and delve into other details in the book germane to ancient history, it might lead the reader astray from the theme. How can the author have the freedom to connect two nodes for the benefit of hypertext browsers without forcing the traversal algorithm to follow that path? A related solution was available years ago in the FRESS system which allowed keywords to be chosen by the user and then determined which paths could be followed and which could not [24]. For this research a modified depth-first traversal was introduced and allowed authors to mark links as non-traversable. Internally, this amounts to another attribute for a link in the frame representation called the 'dead-end' attribute. If the author elects to say that a given link is a dead-end, then the traversal algorithm will not follow that link. For instance, if the author wants one tangential paragraph about the relationship between 'microtext and 'Vannevar Bush' but wants a major section about 'Vannevar Bush' connected to 'macrotext', then the 'microtext' to 'Vannevar Bush' link would be marked as a 'dead-end'.

4.2 Deeper models

Medicine has many well-established models. In medical books one can anticipate that a description of a disease will be decomposed into first a subsection on the etiology, then a subsection on the diagnostic signs, then a subsection on the treatment, and finally a subsection about prognosis. This organization reflects a time sequence and is the basis of the disease model of medicine. In a biology text, one might expect that a section about mammals would have subsections about particular mammals, and the organization of those subsections would reflect that of the introductory section about mammals. In the introduction to the Macrotext chapter of the *Hypertext* book, word-based and indexing language approaches are emphasized. Accordingly, the Principles and Systems sections of the chapter inherit this pattern (see Figure 13). Detecting these patterns and helping them be most noticeable in the book has been facilitated by the programs to traverse the document and programs which give summaries of the characteristics of the node names.

Looking at the top-level outline of two chapters in the book one sees an attempt to model in a consistent fashion concerns in the real world. The only non-identical parts in the top-level outlines of the Microtext and Macrotext chapters are the parts on translations – 'Text to Microtext' and 'Hypertext to Hypertext' (see Figure 11). 'Text to Microtext' is about the semi-automatic connection of text to hypertext and 'Hypertext to Hypertext' is about the semi-automatic connection of one hypertext with another hypertext. These sections naturally follow one from the other, and thus the organization of the headings in these two chapters follows patterns.

```
Macrotext
        Introduction
        Principles
                Word-based Principles
                Indexing Language Principles
        Systems
                Word-based Systems
                Indexing Language Systems
```

Figure 13. Patterns in the Macrotext subsections inherit the attributes of macrotext. (The outline comes from the semantic net but the link labels are not printed here)

4.3 Titles and captions

After the traversal program has determined the order in which links will be crossed, a printing program takes the semantic net and the paragraphs and produces a camera-ready document. The traversal has produced an ordered tree which the printing program labels as an outline. For instance, the first subsection of Chapter 2 is tagged as Section 2.1. Additionally, the name of the tree node is printed as a heading. If Chapter 2 begins with the source node ‘microtext’, then the program actually prints ‘Chapter 2. Microtext’.

In an early version, at the beginning of each paragraph the relevant node-link-node triple was printed. For instance, the paragraph attached to ‘hypertext has history’ would be preceded by that triple in the paper document. After the first draft of the book was distributed to people for criticism, a common complaint was that these triples were annoying. The triples did not add significant meaning to the document and they interfered with the flow of thought. For the authors, the semantic net had been so important that to see node-link-node triples before each paragraph provided a kind of unity to the document. But to outside observers this was not so.

This conflict between what the creators of the system like and what others like is common. Douglas Engelbart’s group published a paper in which each paragraph was prefaced by a number that exactly specified that paragraph’s position in the outline [25]. For instance, the number 1.3.2 would indicate that the paragraph was in section 1, subsection 3, and subsubsection 2. While the original version of Engelbart’s famous paper showed this structure, subsequent reprintings of the paper have removed that feature – this despite the claims by Engelbart that his group found such presentation extremely helpful.

One helpful critic suggested that rather than remove the node-link-node triples that they be placed in the margins. That approach has now been adopted with a twist. Since the node-link-node triples are not as meaningful to readers as they are to those who made the semantic net, the author may place any phrase in the margin. The caption is printed in the margin with the paragraph. While often the caption is the target node name, sometimes an attractive caption does not fit the semantic net paradigm. For instance, if the paragraph associated with ‘word frequency has principle’ emphasizes that ‘frequency times rank equals a constant’, then that equation should be the caption. Fitting that equation into a semantic net would be difficult. In the representation of the extended semantic net in the computer a caption attribute is added.

Printing a section heading as a node name of the semantic net also sometimes seemed inappropriate for the printed form. In the semantic net, each distinct node must have a distinct name. Thus, if there is to be a section about ‘microtext history’ and another section about ‘macrotext history’, they cannot both go by the name ‘history’. Yet, to begin a chapter whose title clearly says ‘Microtext’ with a section called ‘Microtext History’ seems redundant. Given the context it would be enough to tell the reader that he is entering the ‘History’ section. The reader of the linear document will know that this must mean the ‘history of microtext’. The semantic net does not have a particular linear version burnt into it and must distinguish different history subsections by giving them distinct names. Accordingly, the representation of a link object was expanded to include the attribute ‘title’. As the printing program traverses the semantic net and generates headings on paper, it first looks for a ‘title’ attribute. If there is no ‘title’ attribute, then the node name is used. Since the proper title to print depends on the linear context, a more sophisticated system would note the path followed to a node and print the title which suited the context.

4.4 Local cohesion

The greatest criticism which reviewers made of the early drafts of the book was that it lacked cohesiveness. The book read as though it were a collection of notes rather than a single document. Some speculated that this may have been a product of trying to create a hypertext and converting that into text. As one attaches the paragraphs to the semantic net, one cannot be certain from which direction the paragraph will be visited. Accordingly, one is constrained in the way which one can refer to the preceding paragraph in the printed document — because one does not know what the preceding paragraph will be. One might argue that this feature of hypertext will forever prevent it from being convertible into cohesive linear form.

Language is richly composed of many references which set up a commonality of theme between different parts of text or speech. There exist many different types of cohesive tie, but one of the most common is the pronoun. Pronominal substitution is one of the several methods to link sentences and paragraphs and to allow the perception of an overall text. For example:

The batter hit the ball well. It soared into the air.

Within these two sentences, the situation is easily understood and explained. The *it* in the second sentence refers back to *the ball*. The two sentences are bound together by a cohesive tie. By studying the number of occurrences of cohesive ties and their locations, one can begin to gain an indication of how well-formed a text is.

One approach to the hypertext-to-text coherence problem is a labor-intensive one and treats the hypertext form as a rough draft. In the final phase of the document production the hypertext is forgotten, the document becomes exclusively a linear phenomenon, and references to linearly preceding paragraphs may be directly introduced. For instance, if one paragraph discusses the history of World War II and the preceding paragraph discusses the history of World War I, then the second paragraph may be modified to include some introductory sentence such as, ‘The next world war . . .’.

Another approach to the local coherence problem benefits from having the paragraphs on the node-link-node triples rather than on the nodes. Built into the network are transition paragraphs for many possible traversals. For instance, the paragraph connected to the ‘hypertext include microtext’ triple introduces ‘microtext’ from the perspective of ‘hypertext’. The paragraph connected to the ‘principles apply microtext’ triple introduces ‘microtext’ from the perspective of ‘principles’. Thus when the traversal reaches ‘microtext’ from ‘hypertext’, a different story is told, than when the traversal reaches ‘microtext’ from ‘principles’. The traversal algorithm will place the appropriate transition paragraph in the linear document, and a certain, local cohesiveness will be maintained. This approach does not solve the problem of continuity that depends on more than the single preceding node. Furthermore, a dense, hypertext, semantic network, from which a single printed document was to arise, would include many transition paragraphs that would not appear in the printed document.

5 DISCUSSION

In the common approach to hypertext a block of text has a button which corresponds to an unlabeled link from the button to another text block. The approach advocated here has a semantic net whose link objects or ‘node-link-node’ triples point to paragraphs. Several experiences have been described to show how a textbook was prepared on a hypertext system with a semantic net underpinning and through document reuse. The authors are able to grab paragraphs from various sources and index them into the semantic net. Indexed paragraphs are reused in various traversals of the semantic net. The finished book has been automatically generated and typeset directly from the database of paragraphs and link objects.

At times authors may want to express concepts and relations that a semantic net does not support. For instance, one might want to say that ‘for all x there exists a y such that f(x) equals y’, but a semantic net does not support such statements. If, however, one introduces more robust representation schemes, then one may also increase the cognitive load on the author. A semantic net has the virtue of simplicity.

As the semantic net grows, understanding it becomes a challenge. Furthermore, the content of an indexed paragraph is not obvious from its reflection in the semantic net. By having the paragraphs indexed as node-link-node triples, one gets better insight into the relationship between the paragraph and the semantic net than when the paragraph is indexed by just a node name. By placing the paragraphs on the node-link-node triples, the traversal of the semantic net and the generation of cohesive linear documents is also facilitated, as paragraphs are invoked only when the two nodes which they connect are simultaneously considered.

The semantic net has been studied and modified based on principles of analogical inheritance. But while the patterns in the net identified by analogical inheritance have proved useful, simpler patterns have also made a difference. Simply noting how often node names occur and looking for repeating patterns has been useful.

The study of node names suggests generic, semantic attributes for a book about information systems. For instance, any such book might meaningfully have sections on ‘Principles’ and ‘Systems’ and the ‘Principles’ section might be decomposed into subsections on ‘Computer Principles’ and on ‘Human Principles’. Alternatively, these topics might be reflected in a set of link types, such as ‘principles’ and ‘systems’.

In many practical settings, the same body of information has to be included in different documents. The opportunity to see a semantic net and dynamically generate outlines from it may help authors mold the desired document. This kind of interaction is consistently recognized as important for the authoring of documents and seems likely to be useful for document reuse as well.

From the experiences described in this paper requirements for a new hypertext system arose. The ongoing experiments in this research project are taking place on a network of Unix workstations with a prototype document reuse tool that has a relational database back-end and an X-windows interface. The system supports both creation and accessing of hypertext and text. Users can register complex discussions and annotations within the system. Search facilities as well as browse facilities are available. Rough notes may be entered and do not need to be attached to the semantic net. As all entries in the database are tagged with the name of the person who entered them and the date of entry, certain types of retrieval can be done independently of the semantic net. Text-to-hypertext and hypertext-to-text tools have been incorporated in the prototype. The direct-manipulation interface allows users to select a link object and be shown a paragraph or conversely. The outline is dynamically generated, and users can select a topic in the outline and see the paragraphs associated with the corresponding link object. The role of the semantic net is being explored in this new environment.

ACKNOWLEDGEMENTS

The referees have made many valuable suggestions.

REFERENCES

1. Norbert A. Streitz, Jorg Hannemann, and Manfred Thuring, ‘From Ideas and Arguments to Hyperdocuments: Travelling through Activity Spaces’, *Proceedings Hypertext ’89*, New York, pp. 343–364 (1989).
2. John Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, Michigan, 1975.
3. Christine Neuwirth and David Kaufer, ‘The Role of External Representations in the Writing Process: Implications for the Design of Hypertext-based Writing Tools’, *Proceedings Hypertext ’89*, New York, pp. 343–364 (1989).
4. Ruben Prieto-Diaz and Peter Freeman, ‘Classifying Software for Reusability’, *IEEE Software,* **4** (1), 6–16 (1987).
5. Roy Rada and Brian Martin, ‘Augmenting Thesauri for Information Systems’, *ACM Transactions on Office Information Systems*, **5** (4), 378–392 (1987).
6. James M. Nyce and Paul Kahn, ‘Innovation, Pragmaticism, and Technological Continuity: Vannevar Bush’s Memex’, *Journal of the American Society of Information Science,* **40** (3), 214–220 (1989).
7. P. David Stotts and Richard Furuta, ‘Programmable Browsing Semantics in Trellis’, *Proceedings Hypertext ’89*, New York, pp. 27–42 (1989).
8. Jeff Conklin, ‘Hypertext: An Introduction and Survey’, *Computer*, **20** (9), 17–41 (1987).
9. George Collier, ‘Thoth-II: Hypertext with Explicit Semantics’, *Hypertext ’87,* Chapel Hill, North Carolina, pp. 269–289 (1987).
10. Nicole Yankelovich, Norman Meyrowitz, and Andries van Dam, ‘Reading and Writing the Electronic Book’, *Computer* 15–30 (1985).
11. Richard Forsyth and Roy Rada, *Machine Learning: Expert Systems and Information Retrieval,* Ellis Horwood, Chichester, 1986.
12. Roy Rada, ‘Guidelines for Multiple Users Creating Hypertext: SQL and HyperCard Experiments’, in *Computers and Writing: Models and Tools*, ed. Noel Williams, Blackwell/Ablex Publishing, pp. 61–89, 1989.
13. Marvin Minsky, ‘A Framework for Representing Knowledge’, in *The Psychology of Computer Vision*, ed. Patrick Winston, McGraw-Hill, New York, pp. 211–277, 1975.
14. Andrew E. Wessel, *The Implementation of Complex Information Systems,* John Wiley, New York, 1979.
15. Dagobert Soergel, *Indexing Languages and Thesauri: Construction and Maintenance,* John Wiley, New York, 1974.
16. Hafedh Mili, ‘Building and Maintaining Hierarchical Semantic Nets’, *Doctoral Dissertation*, Washington, D.C. (1988).
17. Hafedh Mili and Roy Rada, ‘Inheritance Generalized to Fuzzy Regularity’, *IEEE Transactions on Systems, Man, and Cybernetics* (1990).
18. Roy Rada and Hafedh Mili, ‘A Knowledge-Intensive Learning System for Document Retrieval’, in *Knowledge Reorganization and Machine Learning*, ed. Katharina Morik, Springer-Verlag, New York, pp. 65–87, 1989.
19. Roy Rada, *Hypertext: from Text to Expertext,* McGraw-Hill, London, 1991.

20. Peter Roget, *Roget's International Thesaurus*, ed. Robert Chapman, Thomas Crowell, New York, 1977.
21. Craig Boyle and James Snell, 'Intelligent Navigation for Semistructured Hypertext Documents', in *Hypertext: State of the Art*, ed. C Green, Intellect Limited, Oxford, pp. 28–43, 1990.
22. Jeff Conklin and Michael Begeman, 'gIBIS: A Tool for All Reasons', *Journal of American Society of Information Science*, **40** (3), 200–213 (1989).
23. T. R. Green, 'Pictures of programs and other processes, or how to do things with lines', *Behavior Information and Technology*, **1** (1), 3–36 (1982).
24. Andries van Dam and D. Rice, 'On-Line Text Editing: A Survey', *ACM Computing Surveys*, **3** (3), 93–114 (1971).
25. D. C. Engelbart and W. K. English, 'A Research Center for Augmenting Human Intellect', *Conference Proceedings of the Fall Joint Computer Conference*, **33**, Washington, D.C., pp. 395–410 (1968).