A PEDAGOGICAL ROLE FOR CONCEPTUAL MODELS IN USER-INTERFACE DESIGN

by

Edward Kenneth Brown

A thesis submitted in conformity with the requirements
for the Degree of Doctor of Philosophy,
Graduate Department of Education, in the
University of Toronto

A Pedagogical Role for Conceptual Models in User-Interface Design

Ph.D. Thesis, 1993

by Edward Kenneth Brown

Graduate Department of Education, University of Toronto

Abstract

Conventional user-interface design concentrates on reducing the user's cognitive load with respect to the task they are using the computer to accomplish. In contrast, this thesis identifies a number of *pedagogical roles* in which the user interface can help challenge the student intellectually. Based on Perkins' (1986) constructivist theory of "knowledge-as-design", this analysis suggests that a particular kind of software feature - called an *undetermined* feature- may be beneficial in engaging users in constructive thinking. Specifically, it may provide *extensibility* - the ability for users to construct their own understanding through re-interpretation of screen elements.

A hypertext environment was developed which incorporates undetermined features, without introducing any difficulties in operation of the program. In an empirical study, subjects used iconic anchors to construct links within a hypertext document. A treatment group made use of undetermined anchor symbols to create their own link semantics, while a control group used predetermined link semantics. While the linking task itself reduced the number of facts recalled from the document, the treatment group's retention of recalled facts was improved significantly, by about twenty percent after a one week latency period. Analysis of video protocols from a separate study showed that treatment subjects spent more

time considering the textual material as a whole rather than as separate components.

Additional work is needed to encourage students to explore more alternatives in their semantic designs. Long-term studies, additional software, and instructional support are areas that could help reveal the full potential of an extensible user interface.

The question is not whether machines can think, but whether men do.
- Benjamin F. Skinner

"My mind is working."
- Olive R. Burt

Acknowledgments

Professor Robert. S. McLean supervised this thesis and provided guidance throughout my doctoral program. I will particularly remember his steadying influence during those emotionally stressful times.

I thank professor Rina Cohen for her consistent support. Her suggestions were always tempered with interest and insight.

Professor Mark Chignell added clarity, vision and energy to my research without which I could not have managed.

I also thank the rest of my thesis committee for their efforts: professor Bill Olivier, reader; professor Charles Meadow, extradepartmental examiner; professor Mark Kinnucan, external examiner.

My early efforts in this work were strongly influenced by professors Carl Bereiter and Marlene Scardamalia, and I am grateful for the time and resources they put at my disposal.

My Ph.D. experience was shaped primarily by association with fellow students: Rhona's friendship in particular as well as many others come to mind: Terry, Ken, Dina, Fern, Brian, Jim, Doug, Esther, Hilda, Carol and Janet. From outside my department, the Hyperactives: Deb, Bev, and especially Lisa. Thanks to you all, and to those that have momentarily slipped my mind.

I received financial support from the Ontario Institute for Studies in Education, the Ontario graduate scholarships program, Memorial University of Newfoundland, and a drab from Mark's NSERC grant.

Two department chairs I have served under at Memorial University have been supportive during my period of leave. Thanks to Jane Foltz and Paul Gillard.

Most of all, thanks to Margo for patience and perseverance.

And Jake for his excellent attitude.

Table of Contents

1. Introduction

The use of concrete materials is a pervasive theme in education. The idea that understanding is improved through active participation with the world outside the mind, as a basic observation, is acceptable to a wide variety of educational philosophies.

The computer crosses the boundary between media and concrete material. It can package and present information like presentation media, and can also be responsive and malleable like a real world environment. The question of how to apply these hybrid capabilities is at the heart of the use of computers in education.

Seymour Papert's book *Mindstorms* (Papert, 1980) promotes a particular vision of the computer as an open-ended learning environment. The idea of using a programming language to provide concrete manipulation of the abstract is summarized by his depiction of the turtle as "an object to think with".[1]

This thesis takes seriously the notion of the computer as a new kind of concrete, manipulable substance, and the question of its potential benefits. Of particular interest is the part of the software experienced and manipulated by the user - the "concrete material" part of the software, if you like - conventionally called the *user interface*. The user interface determines how the intellectual content of the computer software is encountered by the student. It is concrete material to the educational software's instructional design.

The idea of the user interface as a material to be manipulated and explained by the user is antithetical to conventional UI design literature and practice. The prevailing notion is that the UI embodies the *designer's conceptual model* (Norman, 1991), which is a static representation of the task components. The user is to acquire this model in order to perform the task efficiently.

In augmenting this approach, this work reviews ideas about the relationship between concrete manipulation and abstract thinking. The emphasis is on examples of existing educational software, although

[1]Papert put it this way: "the computer can concretize (and personalize) the formal... Knowledge that was accessible only through formal processes can now be approached concretely."

implications of non-computer methods are considered as well. Alternatives to a rigid conceptual model are classified into a framework which characterizes existing user interfaces and suggests the introduction of the *extensible* user interface. With an extensible user interface, the users participate in the act of design, interpreting and giving meaning of their own to concrete symbols.

As part of this work, a program was developed to explore the notions of extensibility and student as designer. This chapter traces the origins of these ideas. Evidence of specific effects follows in later chapters.

1.1. Constructivism

> …the object only exists, with regard to knowledge, in its relations with the subject and, if the mind always advances more towards the conquest of things, this is because it organizes experience more actively instead of mimicking, from without, a ready-made reality. The object is not a "known quantity", but a construction (Piaget, 1950, p.375)

The belief that the mind actively constructs knowledge is known as constructivism. A consequence of this belief is that the most effective learning occurs when the learner's understanding is challenged. By Piaget's account, resolving this state of *disequilibrium* requires a constructive effort, or *accommodation* of integrating new knowledge with old.

The radical form of constructivist epistemology divorces knowing from any notion that reality is the referent of knowledge. Knowledge is built up through a succession of subjective experience interpreted though individual beliefs and viewpoint.

This outlook leads to strong criticism of conventional curricula, when it is comprised of a structured sequence of instructional objectives. Such curricula presumes that broad educational goals can be adequately assessed in terms of cumulative success at each sequenced step. From a constructivist viewpoint, this misses the basic tenets of learning: that the learner must be active and self-regulated. Although some conventional curricula stress active involvement though learner control and feedback, the responses and performance criteria are generally determined by the curriculum, and built into the instructional unit (Fosnot, 1984). They are active, but not self-regulated.

1.1.1. The social solution

Creating alternatives to a conventional curriculum often means changing the conventional classroom model as well. Constructivists favor experiential, self-directed learning environments in which students encounter new "constructive" challenges.

In the CSILE project, the term "knowledge-building" is used to describe the generation of new knowledge through students' interaction and co-operative construction of a public database (Scardamalia & Bereiter, 1991). The focus of the project is on uncovering the skills and tools that best promote learning in such an environment.

Reciprocal teaching (Brown & Palincsar, 1984) also promotes the idea of students learning through active discourse and presentation of their knowledge. The exchange of student-teacher roles put the student in the position of communicating their knowledge. The understanding they possess becomes part of the reflective process of instructing another.

Cognitive apprenticeship (Brown, Collins, & Duguid, 1989) proposes that knowledge is interpreted and achieves meaning only through interaction with real world events and objects. This theory also promotes the idea of bringing manipulation of real world events and objects into the classroom in a manner relevant to individual experience.

These examples demonstrate the range of suggestions that relate to the student as actively involved with construction of their knowledge. These involve changing not only the social character of the classroom, but also the values by which the curriculum is judged. This thesis has little to say about classroom organization, but does provide a constructivist vision of the use of computer software.

1.1.2. Individual effort

Radical change is not a necessary consequence of a constructivist approach to education. Constructivism as a broad theoretical base must also explain how people learn from private study, rote methods, and other "traditional" classroom activities. As far as they change the knowledge of the organism, these activities must also engage a constructive process of some kind.

The problem with self-regulated learning is that it takes a lot of time. Jerome Bruner, the progenitor of the discovery learning movement, has this to say:

> It seems to me highly unlikely that given the centrality of culture in a man's adaptation to his environment… biologically speaking, one would expect each organism to rediscover the totality of its own culture… Moreover, it seems equally unlikely, given the nature of man's dependency as a creature, that this long period of dependency characteristic of our species was designed for the most inefficient technique possible for regaining what has been gathered over a long period of time, i.e.: discovery… (Bruner, 1966, p.101)

In the same sense that concrete materials may be frustrating to someone who can reason without them, social exchange may be distracting for an individual who has independent study skills.

With the potential to make the abstract concrete, the computer provides an alternative to social interaction as a way of making abstract knowledge tangible and a target for reflective thinking. Instead of discussing ideas and information with other people, the learner may actively manipulate a representation of knowledge. Paper and pencil representations of knowledge are ubiquitous. The agenda here is to explore how software could allow people to manipulate knowledge in new or beneficial ways.

1.1.3. Knowledge as Design

The ideas of active use of knowledge and concrete manipulation are neatly encapsulated by Perkins' (1986) theory of *Knowledge as Design.* According to this theory, knowledge acquisition in any form can be understood as an act of design. Even apparently passive activities such as reading are most beneficial when the purpose and structure of the prose (i.e., its design) is reconstructed by the reader. The elegant thing about this theory is that it captures the idea of embedding abstract thinking in the manipulation of concrete materials with a single conventional concept: design.

The advantage of a theory of knowledge as design is that it projects the active involvement of the student onto the objects and events themselves. This provides a means of looking for potential learning activities in the characteristics of the materials. In Perkins' words, one looks for "opportunities for design". (p.98)

The idea of creating design opportunities partly resolves what kind of software fits the constructivist vision. The ensuing question is how to create effective design opportunities with computer software.

1.2. The Research Question

To the extent that the user interface merely prescribes a particular way of understanding and performing a task, it demands little constructive effort. Manipulating the interface might directly enhance learning if such manipulation involved consideration of what is being represented and how it is represented; in essence, design acts. As an empirical question, the issue is: how can the user interface improve design opportunities?

A complete response to this question would identify qualities or properties of the user interface which improve the value of the software for constructive activity. Unfortunately, there are no general prescriptions or theories to predict what interface techniques or design tasks would elicit such improvement. To operationalize the question, appropriate techniques and tasks must be developed.

Following Perkin's advice, one can look for design opportunities; in this case, looking in the accumulated knowledge of user interface design and educational software that is available. This is the function of the literature review of chapter two. It identifies properties and provides guidelines for improving design opportunities which can be applied and tested in software. If these opportunities are effective, they should increase constructive activity on the part of the student.

In this thesis, the practical expression of these ideas is an extensible hypertext system. Hypertext links can add value to a conventional document in a way that is not possible without a computer (cf. Kozma, 1991). The user interface developed here includes features which (according to this author's claims) should support the constructive aspects of authoring hypertext documents.

The experiments attempt to demonstrate that these features do indeed improve constructive activity. One feature in particular is examined in the context of a hypertext linking task. If this feature actually produces more thoughtful, constructive effort on the part of subjects, it should affect what they learn. Free recall is used to assess whether exposing subjects to this "improvement" results in better retention of the hypertext document.[2] An additional study uses verbal reports to determine how constructive activity is affected by the treatment.

The results presented in this thesis partly resolve the research question. UI guidelines are developed, and their application is demonstrated. A specific user interface feature is demonstrated, with some positive results. While this is as much an approach as it is an answer to the question, the door has been opened for the development and evaluation of user interface techniques which indeed provide design opportunities.

1.3. Concrete Manipulation of the Abstract

In this section several examples of the use of graphical manipulation as a design activity are considered. This clarifies how the manipulable nature of the user interface plays a part in addressing the research question.

1.3.1. Designing to Learn

Design is not a totally original learning activity. Many activity and project based instructional approaches could be characterized as design opportunities. Fewer suggestions apply concrete manipulation broadly to abstract knowledge in general.

Rudolph Arnheim, in his book *Visual Thinking* (Arnheim, 1969), proposes such a role for graphic design. Expanding on the theme of the visual perceptual channel as a primary means of understanding the world, Arnheim explains how graphic design can be used as a basis for understanding geography, biology, mathematics, physics, art, and other basic scholastic subjects.

One current area of investigation is teaching students to draw semantic networks as a means of reflecting on knowledge (Holley & Dansereau, 1984b). Once again, the activity of producing a concrete representation of knowledge is the key to the effectiveness of this strategy. Clarke (1990) reviews a number of graphic organizers which can be used by students for a variety of purposes: mapping expository text, charting processes, defining cause-and-effect relationships, representing class interaction, and so on.

---

[2] Additional effort may consume more time, so subjects may not learn as much of the document, but if the added effort is beneficial, their retention of what they have learned should be better.

1.3.2. Forms and Representation

The common element of these approaches is providing a graphic form as a focus for reflection. The actual form itself may not be of as much interest as thinking entailed by creation of the graphic:

> ...it seems possible that spatial strategies are effective not because they provide an image, but rather because in constructing a graphic representation the learner carries out activities such as analysis, encoding, and organization that are themselves effective regardless of whether or not they result in a spatial representation. If so, the critical elements to be taught may not be the spatial scheme itself but rather the steps taken in constructing a spatial representation... (Goetz, 1984, p. 304)

Transforming information from one formalism to another has been proposed as a valuable learning experience which can be particularly well supported by educational software (Dickson, 1985). The SemNet project (Fisher, 1988) provides the student with semantic-network software, removing much of the mundane mechanical effort from creating and maintaining the network.

Such form-filling activities miss much of the "design opportunity" of creating spatial representations. Clarke (1990) puts it this way:

> Whereas conventions of sentence structure make most writing linear in form, graphic organizers take their form from the presumed structure of relationships among ideas. Different kinds of relationships- like parts to the whole, cause to effect, or evidence to conclusion- suggest different kinds of graphic organizers... The struggle for form is a struggle for control over an idea. (Clarke, 1990, p. 30)

Rather than develop automatic skills for filling out a predetermined form, the student should think about the appropriate representation for each particular situation. The "struggle for form" is a design opportunity that should be achievable in software environments.

1.3.3. Programming as Design

Harel (Harel & Papert, 1990) investigated programming as a design experience using LOGO with fourth-grade students. The design focus was explicit, with the concepts of instructional design and educational software being introduced to the students, as well as specific design strategies being endorsed, such as the use of Designer's Notebooks for students to record their ideas.

The constructivist approach driving the project gave pre-eminent importance to manipulating objects and to concrete thinking.[3] This entails a "correction" of Piaget's notion of concrete operations from a developmental stage to a form of reasoning: "[Piaget's] major weakness is his resistance to giving up the value system that places formal thinking "on top". This resistance leads him to see concrete thinking as child's thinking, and so keeps him from appreciating the full breadth of his discovery of the "concrete" as a universal form of human reason. The… correction can be stated as recognizing that the "formal" is not a stage but a *style* of thinking". (Papert, 1990, p. 6)

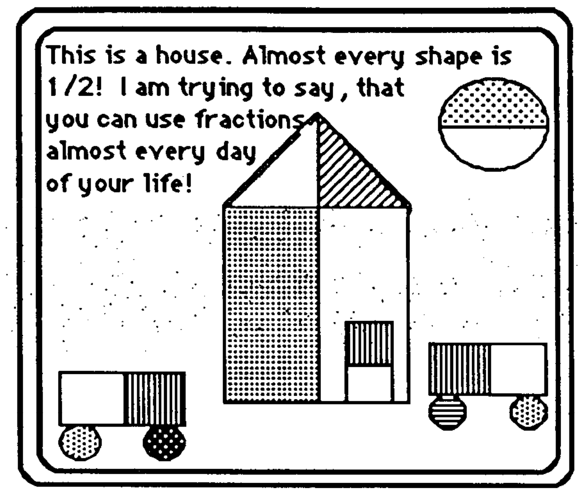


Figure 1.1. from (Harel & Papert, 1990) A fourth-grader's screen on fractions demonstrates careful thinking about the topic, but why force her to learn LOGO to do this?

What is not clear in this case is the contribution of programming and LOGO to the students' design effort. Consider one student's design effort shown in figure 1.1. It is hard to believe that acquiring programming skills, debugging a LOGO program, or even using a computer contributed substantially to the creation of this design. There may indeed be learning

[3] This work comes out of Papert's Epistemology and Learning group at MIT's MediaLab, who designate this emphasis of the concrete as a "new theory of construc*tion*ism". They further promote self-directed activity as the "correct" pedagogy: "When children are responding to questions and exercises in workbooks, one of the things they are learning - regardless of the content of the workbook - is the very idea that learning itself consists of answering other people's questions rather than your own. This model of learning is ultimately harmful…" (Falbel, 1990, p.424). Such an extreme view is not necessary in advocating the potential of programming as a constructive design activity.

value in designing a computer program, but such examples are not convincing in this regard.

In providing a design environment, this thesis avoids placing extraneous burdens on the user, such as programming or even learning to program. Instead, the user interface strives to evoke thoughtful design effort without a prerequisite of special skills. The suggestions in this thesis are in part an attempt to produce thoughtful design effort without the need for a lot of extra preparation or special skills.

1.4. The Extensible Interface

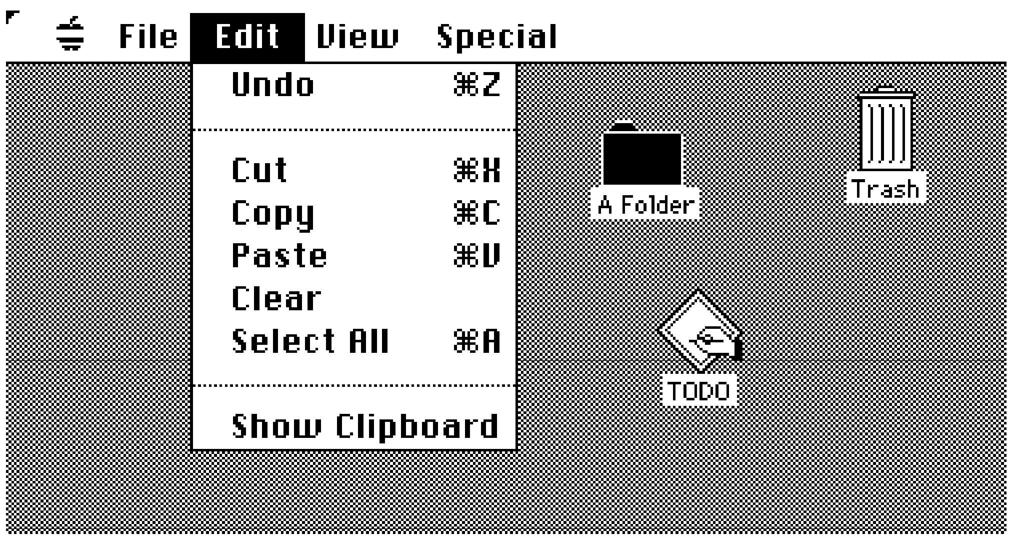


Figure 1.2. The perceptual features of the Macintosh[4] desktop have their meanings prescribed by the designer's conceptual model. The menus, icons, windows, and so on convey the concepts imbedded in the design.

This thesis contends that simply using the designer's conceptual model is akin to "form-filling". To invoke a "struggle for form", the learner should be able to examine their own concepts and create new representations of their ideas within the context of manipulating the interface. In a sense, this author wants the software user to take on something conventionally reserved for the software developer - namely, imparting meaning to the objects and features that appear on the computer screen.[5]

---

[4]Macintosh is a registered trademark of Apple Computer, Inc.

[5]Confusion may arise between the design activity of creating software, and the activity of the user as discussed under the knowledge-as-design theory. For this reason, in the remainder of this document, the term "development" is adopted for the activity of the

The word *extensible* is adopted for a user interface with which the user can introduce and represent new concepts. The user *extends* the range of concepts included in the exchange between human and machine. If the user has difficulty interpreting the interface components in anything other than a fixed and prescribed way, then it is not extensible.

1.4.1. Techniques for an Extensible Interface

The obvious way to provide extensibility is through programming. While providing the requisite flexibility, programming burdens the user with special skill demands. Papert's HeadLight project and diSessa's BOXER software are discussed elsewhere in this regard.

In this work, extensibility is provided without programming. This is accomplished by users simply generating conventions as they operate the interface. The user selects perceptual features of the interface to represent the concepts he/she is interested in.

Many programs have enough flexibility to support such conventions. Any word processor, for example, will provide extensibility simply by virtue of allowing arbitrary text to be entered. The user can decide to use certain text characteristics to represent particular concepts. For example, one could embed the character string "!*!" in the text to indicate important ideas. Furthermore, the user can continue adding such new concepts ad infinitum, simply inventing a new convention each time.[6]

A drawing program would support similar conventions, with different images set aside for different purposes. The commonplace desktop metaphor (figure 1.3, below) has a prescribed use for screen icons, but color and position of these icons is somewhat free.

Perceptual features which allow such conventions are called *under-determined*; they have an intended use in the developer's conceptual model, but their specific arrangement is open to interpretation by the user. Icons on the desktop require different positions in order to distinguish and identify them easily, but their exact position is not determined. Arbitrary

---

software creator. Thus, "user interface development" is substituted for the more conventional "user interface design", and "software developer's conceptual model" for "designer's conceptual model".

[6]Programmers use such conventions all the time. One common convention, for example, is to represent global variables in upper-case letters.

text can be entered in a word processor because that is the nature of text, but the meaning of each character sequence is not part of the conceptual model of the UI. One possible approach to providing extensibility is to deliberately under-determine the way features can be combined and arranged.

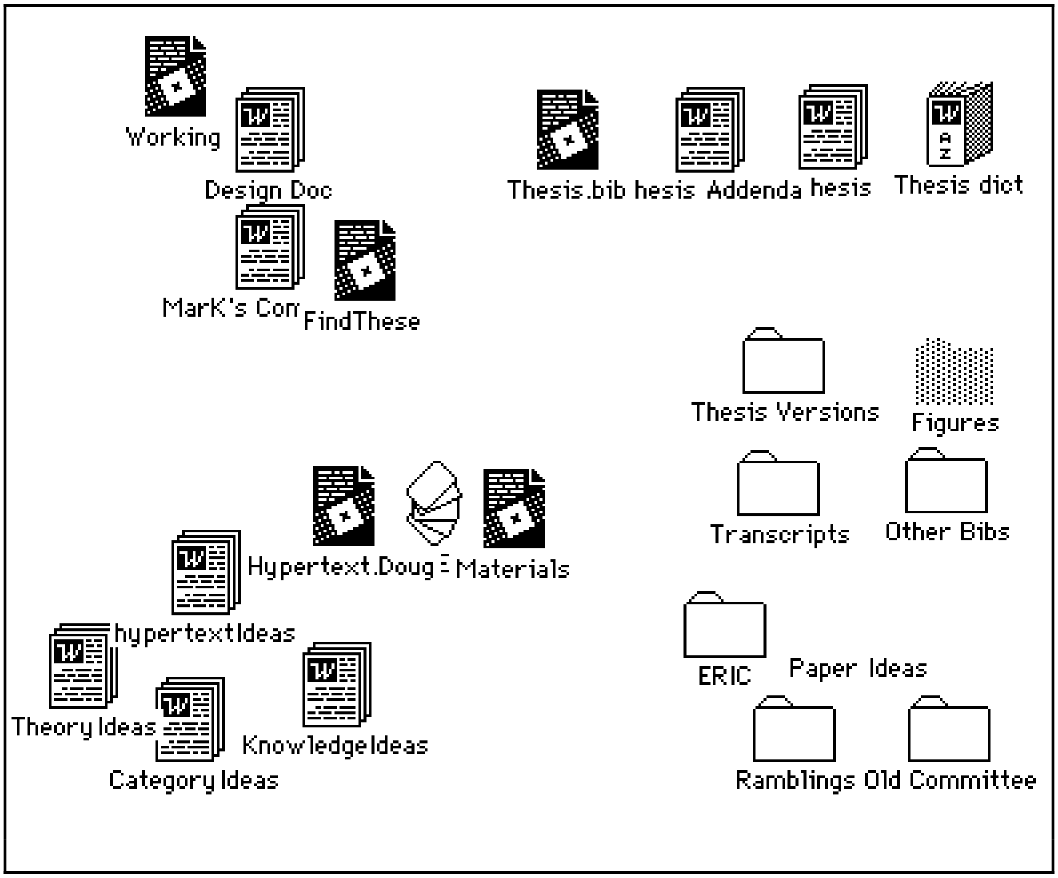


Figure 1.3. The positioning of icons on the Macintosh desktop provides some degree of extensibility

An additional possibility is to include perceptual features which are completely *undetermined*. In other words, the UI would include specific perceptual dimensions which have no prescribed meaning. For example, one might add sounds to a desktop interface, and give the user the ability to assign sounds to different files. The user could then assign the sounds to indicate the attributes s/he felt were of interest.[7]

Deliberate incorporation of underdetermined and undetermined features is an original suggestion. One purpose of this thesis is to explore the

[7]In fact, the Macintosh desktop allows the user to assign colours to different files. However, there are only eight colours available for whatever the users want to indicate.

potential of such features. Will users spontaneously exploit extensibility? Are some forms of extensibility better than others? What would make one interface more extensible than another? What are the implications for education? These questions open a new avenue for research. With respect to the original research question, the important issue is: can underdetermined features be exploited in the users' design acts?

1.5. Summary

In considering the potential of the user interface as an educational material, the constructivist outlook suggests the development of software that provides opportunities for the student to learn through design activities. Undetermined UI features could produce an extensible interface which allows the user to represent original ideas, without making complex skill demands on the user.

In chapter two, a review of literature and existing software helps define the context in which these proposals arise. By classifying different pedagogical uses, or "roles" for software, it is possible to prescribe different UI characteristics, including, when applicable, extensibility.

Chapter three discusses the development of software and experiment to study specific UI characteristics. Chapter four presents results, and chapter five discusses the success and implications of these efforts.

2. Literature Review and Theoretical Framework

This literature review examines how educational prerogatives are taken into account in the creation of user interfaces, from a knowledge-as-design perspective. It begins with a critique of current ideas in the field of human-computer interaction, and moves on to an examination of educational software which reaches beyond those ideas.

Different ways of exploiting the user interface are identified and discussed through the use of specific examples. This coincides with a framework for matching desirable characteristics, such as extensibility, with the pedagogical role of the software.

Later sections of this chapter discuss the particular interest in hypertext as a constructive environment, and its merit for examining the practicality of these ideas, leading to the experiments described in chapter three.

2.1. The user interface

> How shall the scientist grasp the virtual objects? And how, by grasping objects, grasp concepts as well? (Brooks, 1988)

The development of a computer program includes the creation of a user interface. This interface defines the manner in which the user and computer interact, and ultimately determines how the user perceives and understands the activity or task being performed with computer support.

Unfortunately, few educational applications incorporate educational priorities at the level of the user interface. Instead, educational objectives are used to prescribe particular tasks or activities which are then fit to commonly known user interface paradigms.

A typical educator or instructional designer will not have the expertise to create a new style of interaction for their specific needs. At best, such a person will draw on accumulated experience with educational software for their ideas concerning the user interface. On the other hand, if a user interface expert is brought into the project, the expert will apply the standard rationale of production software. The user interface will be intended to make the program easy to use, without reference to the educational goals of the project.

The current practice for user interfaces concentrates on the appropriateness of the software; first, to the task and second, to its users. A good analogy is a custom-designed set of golf clubs. They are suited for the game, but also fitted to the individual's characteristics.

From a knowledge as design perspective, this is an inadequate approach. It is not enough to provide new ways to perform, or even perceive, a task. To effectively change a person, or change what the person knows, the person must undertake a design effort of some sort. Fitting the user interface to what the person already knows effectively eliminates the opportunity to contribute to such design effort.

2.2. Educational Interfaces

An interface design may not always be recognized for its contribution to the success of a program. A case in point is the GEOMETRY tutor (Anderson, Boyle, & Yost, 1985). This software is presented as a successful application of ACT* theory[8] to the design of an artificial intelligence tutoring program. However, many subjects reported that the proof graph display (i.e. an interface component) was the most helpful part of the program. This graphic was not developed as a theoretical concern, so the success of this program may owe as much to the particular interface as it does to ACT* theory.

A dramatic example of the impact the user interface can have on student's understanding of a conceptual model is described in Cohen & Geva (1989). Young students' difficulty with spatial concepts in the LOGO programming environment are successfully addressed by modifications to the LOGO interface. Some of the modifications are both simple and subtle, such as requiring the Turtle to make its steps visually distinct (p.353).

Such a solution compares very favorably to instructional remediation on children's spatial concepts, as suggested in Thompson (1987). The sophisticated artificial intelligence program proposed to perform such remediation makes for an ambitious project, using an unproven approach. Simple and effective changes to the UI provide a much better remedy.

In the preceding example researchers modified the user interface to provide support for children's understanding of certain concepts. This thesis proposes that the interface should be created in the first place with an appropriate conceptual orientation. The effort here will attempt to

[8] Adaptive Control of Thought is Anderson's general theory of cognition, which has a strong emphasis on skill acquisition.

determine what pedagogical concerns can be addressed through the user interface.

2.3. Different roles for conceptual models

This review concentrates on how the user interface can be exploited for pedagogical purposes. To understand any program, the user must learn to interpret the interface components, to give meaning to the objects on the screen. This is essentially the process of exploring the conceptual model (cf. Norman, 1983) upon which the software developers based the interface.

This process of exploring the conceptual model can be exploited in different ways. In this regard, this review identifies two important characteristics of how the conceptual model is exploited. The first contrasts conceptual models which are familiar to the user with those intended to be new and unknown. A familiar model makes the software easy to use, whereas an unfamiliar one challenges the user to deal with new ideas.

The second contrast distinguishes programs that provide screen objects for specific concepts from those that provide mechanisms for the user to build representations of their own choosing. The added opportunity for the user to create their own designs means the software cannot reflect specific domain concepts in the interface. These two contrasts comprise the framework of figure 2.1.

contrast 2: representation of concepts in interface	*contrast 1:* users' understanding of conceptual model	
	familiar	challenge
fixed in software	**ARTIFACT**	**KNOWLEDGE**
built by user	**FORMALISM**	**MEDIUM**

Figure 2.1. Four roles for conceptual models.

This framework gives names to four roles identified by the two contrasts. These roles are distinct types of conceptual models which support learning differently. A short description of each role follows:

Role 1: Artifact In this role, the conceptual model provides a mechanism or apparatus for presenting content or performing a task. The software "artifact" should be easy to understand, so that the student can spend his/her mental effort on the task of real pedagogical interest.

Role 2: Knowledge The conceptual model is the actual knowledge of pedagogical interest. Constructing an understanding how the software works (or, more accurately, what the interface elements mean) is equivalent to learning the intended domain knowledge. Educational simulations are the prototype for this role.

Role 3: Formalism. A formalism is a structure or formal system for representing information. Under this kind of conceptual model, the user's task is to transform knowledge into a conventional representative form. The software might be considered as a notation method.

Role 4: Medium. A medium provides an artist with creative freedom; the form of expression is their own. Conceptual models in this category have the same perspective; the user is expected to explore different means of representing and understanding knowledge with the software system.

In the following sections of this chapter, literature and software motivating the contrasts of figure 2.1 are discussed, with reference to specific examples.

2.4. The Artifact Paradigm

The theme of the UI providing the user with a conceptual model with which to discern his/her task is well developed in the field of human-computer interaction. A current issue is how software changes not only people's performance of a task, but also their perception and understanding of that task. The term "cognitive artifact" is used by Don Norman (Norman, 1991) to express this notion.

This perspective can be attributed historically to the possibilities that unfolded from the *graphical* user interface. Two key notions that are widely accepted as seminal are the use of *analogy* or *metaphor* and the idea of *direct manipulation* in the interface.

The goal of assisting the user in acquiring the "correct" conceptual model of the computer program (in order to use it correctly) can be powerfully supported by a graphical interface whose appearance and behaviour draws on familiar "real world" objects and processes - that is, a

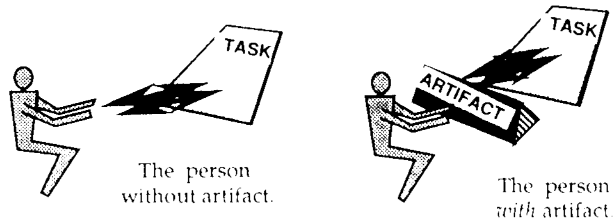


Figure 2.2.1 The system view of a cognitive artifact. Under this view, we see the entire system composed of the person, the task, and the artifact. Seen from this perspective, the artifact enhances cognition, for with the aid of the artifact, a system can accomplish more than without the artifact.

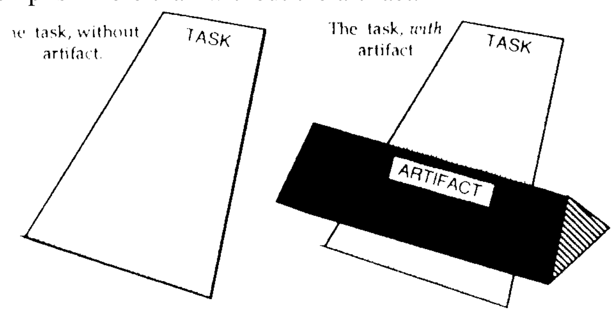


Figure 2.2.2. The personal view of a cognitive artifact. Under this view, that of the individual person who must use the artifact, the view of the task has changed: thus the artifact does not enhance cognition - it changes the task. New things have to be learned, and old procedures and information may no longer be required: The person's cognitive abilities are unchanged.

Figure 2.2 The user interface as cognitive artifact. From Norman (1991)

---

real-world metaphor. The desktop metaphor, as developed in the Xerox Star and popularized in the Apple Macintosh® computer, is the best known example of this principle (figure 1.2). The elements of the interface correspond roughly to objects on a desktop - files, documents, and so on. This metaphor gives the user clues as to what actions may be appropriate and what results might be expected from such actions.

The notion of direct manipulation (Shneiderman, 1983) is that there is no feeling of mediation between the user and the objects of interest in the interface. If, for example, the user learns a command language in order to operate a database, the command language is an intermediary between the user's intentions and the data to be manipulated. The user must express those intentions in an indirect manner. Graphical interfaces and pointing devices allow the user expressing him/herself to perform actions in a "direct" manner: by pointing at and manipulating visual entities on the screen. In fact there *is* mediation in these graphical interfaces, since the user must learn how to manipulate the graphical entities using the input and output devices. However, these mechanisms are simple and obvious enough that they quickly become automatic to the user; s/he feels as though s/he is directly manipulating the objects of interest.

These attempts to capture usability depict a shift in user interface concerns. From the initial prospect of designing a dialogue between man and machine, a new "artifact" paradigm emerges, in which the software changes the way a user perceives his/her task.

The outlook of changing how users perceive a task suggests the potential of the user interface. Under the artifact paradigm, this potential is dedicated to usability. The quest to exploit this potential for educational purposes begins there.

2.4.1. Usability

Usability and learnability of the program are universally promoted in the HCI literature as priorities of the interface development effort. This focus is expressed in the popular catechism "user friendly". Typical principles or guidelines listed in tomes such as Shneiderman (1987) emphasize such things as consistency and obviousness of the interface to promote usability.

The constructivist approach demands that educational software should provide design opportunities. This calls into question the use of a conceptual model merely to make the program easy to use. The appropriate consideration is not whether the software is itself usable or learnable but whether it is a good device *with which* to learn.

It could be argued that this issue is subsumed under the general notion of usability, and therefore is not a different emphasis at all. Usability, after all, is judged by how well the software supports the particular task in question. The problem with this argument is that the concept of task conventionally used in HCI does not encompass the kind of broad and loosely defined goals often used in education. Certainly it does not identify design opportunities to be addressed in different ways by different individuals. Instead, a particular task is isolated, and objectives are specified during the creation of the interface.[9] The constructivist approach

[9]The identification of "task" in UI development is critical. The "task" has to be defined as something that is susceptible to classical "task analysis". This leaves little room for alternatives such as users discovering new problems and setting about to solve them. Greif

rejects the notion of task as something which can be given a fixed, universal interpretation for all users.

UI development methodologies, which structure the process of creating the user interface, reflect this concept of the user's task. As Baecker & Buxton (1987, chapter 11) summarize, the conceptual model of the system (and the manner in which it supports the task) is defined early in the software project. A successful user interface is one which helps the user acquire this conceptual model.

Such a methodology fits in well with an accretion model of learning- knowledge to be delivered in pre-defined units and acquired as quickly as possible. Rote drill-and-practice programs, for example, would probably obtain a quite effective UI under this kind of software development methodology. However, with a knowledge-as-design perspective, one is compelled to ask questions such as "what design opportunities have been provided for the user?" or "which UI techniques will promote conceptual change?"

This does not mean programs built with these methods cannot be used for discovery learning or to promote cognitive objectives. It does mean that the UI will not be directly involved in these processes. In this thesis, it is shown how the user interface may directly support cognitive objectives, such as getting students to think more carefully. Seeking such a role for software takes us beyond the artifact paradigm, where usability and learnability are less important than design opportunities.

2.4.2. HCI beyond the artifact paradigm

Even within the usability agenda, the way in which software artifacts embody an interpretation of the world has become an important theme in human-computer interaction. (cf. Carroll & Kellogg, 1989). For example, it is recognized that the conceptual model not only provides the user with a conceptual orientation with regard to how to use the finished software product (Carroll, Mack, & Kellogg, 1988), but also provides the software creators with ideas and objectives throughout the process of developing an interface. (MacLean, Bellotti, Young, & Moran, 1991).

(1991) shows task analysis is inadequate when people have even as few as two alternative ways of interpreting a task. Thus, the identification of "task" by the designer inherently limits what conceptualization of the activity the UI will support.

Because they are interpretations, interfaces based on metaphors can be problematic. When a metaphor is too simplistic, misconceptions and errors can result (Cardinale, 1991). The alternative is a conceptual model derived from the understanding of experts. The expert understanding approach produces a more difficult interface, but avoids naive expectations of the software.

This wider perspective has been accompanied by new technologies and new ideas for interfaces. The notion of enhancing an image by adding visual cues (drawn from the conceptual model) is termed *visualization*. (e.g. Brooks, 1988; Rueff, 1992). The additional conceptual presentation can be valuable even when the actual phenomenon is directly available for study. For example, air in a wind tunnel can be rendered as an opaque fluid on a graphics screen. *Virtual reality* goes beyond graphics presentation to immerse the user in a computer generated environment, complete with a 3-D visual field and multi-sensory feedback.[10] *Force-feedback* manipulators add the possibility of haptic output; chemists actually feel the strength of chemical bonds as they handle giant simulated molecules.

Brooks (1988) goes as far as to suggest that computer generated interactive models will replace mathematical models as the principal tool of the physical sciences. The ability to interact personally with these models is a powerful new way of understanding - as evidenced by chemical artificial worlds which have led to the discovery of new molecular structures.

Simple artifacts do not have to behave like real world entities; they merely have to help get a job done. These interactive models, however, must be faithful to the real world entities they represent. That is, they embody knowledge which could be used for educational purposes.

2.5. The conceptual model as knowledge

There are important distinctions between a metaphorical world designed for usability purposes, and a simulation which truthfully presents a real-world phenomenon. This is the difference between the artifact and knowledge roles of figure 2.1.

---

[10]the equipment usually includes at a minimum: one output screen covering each eye, providing stereoscopic images, earphones, and a device for sensing body movement, such as a dataglove. Some setups are much more complex.

With simulation software, students can explore and manipulate environments which demonstrate principles of chemistry, physics, economics or other subjects of interest. Once the student has learned all the concepts in the simulation software, its educational value is exhausted. This is in contrast to the artifact paradigm, in which learning the software is merely a hurdle *before* performing the task of interest.[11]

Since these programs simulate real world properties, they are constrained in the kind of behaviour the interface might exhibit. In other words, the "metaphor" is not primarily a mechanism for usability, but carries much of the instructional content.[12] From here, this author infers that in such a situation, the main design consideration is not usability, but truthfulness of the interface.

The work of Hollan, Hutchins, & Weitzman (1984) anticipates this departure from the artifact paradigm. Their interactive simulation of a propulsion plant is based on the mental model they want the student to adopt. They make a deliberate choice to render a view of a propulsion plant which is conceptually accurate rather than physically accurate or simple to operate. They consider the running simulation to be a "continuous explanation" of a conceptual model. The ability of the software to render a faithful conceptual view of the process under study is termed *conceptual fidelity*. In contrast to the direct manipulation of a physical metaphor, the goal of their software is for the users to learn the important constituent concepts of the simulation[13].

2.5.1. A different target system

Conceptual fidelity suggests a different intent for the conceptual model.

In fact, the essential notion of conceptual model is broad: "A conceptual model is invented to provide an appropriate representation of the target

---

[11]Note how simulations are encompassed by the artifact paradigm in their normal commercial use: when the simulation is used as an analytical tool for particular data. In educational setting, however, they are more often used as described above; for teaching the conceptual model embedded in the software.

[12]A real simulation can be considered metaphorical only in a particular sense: the properties of the real world which are simulated are those that the software builders thought to be important.

[13] As Papert (1980) points out, the conceptual model not only isolates particular aspects of real-world phenomenon, but eliminates confounding information that does not play a role in the model. For example, a simulation of Newtonian physics might eliminate friction and gravity from the model, to make other concepts more apparent.

system, appropriate in the sense of being accurate, consistent, and complete. Conceptual models are invented by teachers, designers, scientists, and engineers" (Norman, 1983,p. 7).

It is the HCI literature which assumes the target system to be modelled is the software or computer system (e.g. Cardinale, 1991). The conceptual model is considered a vehicle for conveying how to use the software.

For teaching purposes, the "target system" may be something other than software. As a hypothetical example, suppose the topic of instruction (i.e. the "target system") is orbital mechanics. The conceptual model might be a model of planetary motion. If a computer simulation is used, then it must adhere to the appropriate conceptual model.[14] The user interface developer does not have the luxury of changing this model to make the software easier to use.

Students studying this simulation would not be learning the conceptual model in order to apply the software to a subsequent task. Rather, they are acquiring the conceptual model for its own inherent value. Instead of offering a familiar metaphor, the interface presents a challenge of new understanding.

2.5.2. Thinkertools: simulation software

Thinkertools (White, 1988) is an example of the conceptual model in the knowledge role. The topic is Newtonian physics. A simple visualization technique displays a vector representation of forces acting on an object as well as a representative view of the actual motion of the object. The elements of the conceptual model (i.e., the force vectors) are represented graphically, and can be inspected as they change over time.

Thinkertools is one component of a larger physics curriculum, involving other kinds of experimentation and classroom interaction. This larger perspective clearly places the software in the knowledge quadrant of figure 2.1. White has avoided the common trap of cramming as much of the

[14] One might argue that there are in fact two "target systems" here: the academic topic, and the software. However, separating the two corresponding conceptual models would be difficult, if not futile. Certainly the academic topic should inform the software design and have a definite impact on that system. If one insists on thinking of two separate conceptual models, then this thesis deals with the academic one, and how it should be applied in the design of software.

curriculum as possible into the computer. Instead, her research identifies a specific role for the computer which it performs beautifully.

Although visualization techniques and simulation type software might suit the knowledge role well, they neither define nor restrict that role. diSessa (1982) suggests a visualization approach as a general strategy for helping students use formal symbol systems. He adopts the term "semi-formalism" to describe runnable models of abstract reasoning. The idea is that such graphic presentations would help the student acquire a basic understanding of formal reasoning in general. A similar analysis is offered by Tiessen (1991) who relates Bruner's three modes of thinking to the computer's ability to reify processes graphically. By tying a symbolic system to enactive manipulation via a graphical user interface, a middle ground is created for comprehension; i.e. an iconic mode for using formalisms.[15]

Under the knowledge role, the software presents a carefully chosen representation of some domain. This representation is such that it makes the domain knowledge more comprehensible. In the next section the prospect of a *general* representational system, which can represent knowledge of any domain, is discussed.

2.6. The conceptual model as formalism

The student can be provided a great deal of design latitude through software that allows the user to build their own representations. Given UI components that reflect structures in a representational *system*, the user can depict concepts from different topics of study.

SemNet (Fisher, 1988) offers an example of a formal structured mechanism which *does not* provide special representations for specific domain concepts. It is essentially a study tool for creating and maintaining semantic networks via computer. The user labels ovals with domain concepts, and arcs with the relationships connecting those domain concepts.

In exchange for the power to span different topics, such formalisms lack the ability to distinguish specific topical concepts in unique ways. Domain concepts such as force, gravity, inertia, and so on simply do not appear in

[15]Alan Kay (1987) also uses Bruner's modes, but his objective is to make the interface more comprehensive, not to help develop thinking.

the conceptual model, so they do not appear on the screen. In contrast, Thinkertools displays force vectors which behave according to the principles of physics. Thinkertools cannot be applied to other topics, even science topics such as chemistry. The capability to express chemical concepts simply doesn't exist in its conceptual model. On the other hand, this software can represent the concept of force in a particularly cogent way.[16]

While it may not provide the best representation for any one domain, the premise of the formalism role is that the design act of translating knowledge into a new form will be a powerful learning event. The most comprehensive work on having students perform this kind of translation relates to paper formalisms, called *graphic organizers*.

2.6.1. Graphic organizer formalisms

As a student's study aid, graphic organizers are usually promoted through the ideas of Ausubel (1963), in terms of meaningful learning. The idea that connecting new concepts to old is essential to meaningful learning leads to the practice of constructing a physical representation of such connections as a learning device.

Most of the graphical organizer devices take the form of a semantic or propositional network. Quillian (1968) was one of the first to suggest human memory could be modelled as a network of idea "nodes" and relationship "links" between nodes. Such networks have become the most successful model of human memory (Holley & Dansereau, 1984a), prompting the use of such spatial representations as learning devices. The rationale is simply that if the representation is parsimonious with human memory organization, it will be easy to learn.[17]

Graphical mapping techniques developed from a learning theory perspective are usually part of a larger *spatial learning strategy*. The actual activity of using or creating the specific graphic is one stage of a strategy that may involve preparation, review, self-monitoring, and other processes that are seen as essential to the success of the study method.

---

[16] This has implications other than the user interface. Only when the domain is fixed can domain specific techniques be employed, such as using artificial intelligence to model the domain knowledge.

[17] Naturally, there are objections to this type of reasoning; e.g. Holley et al (1984c) p.25.

Goetz (1984) summarizes the activities involved in a spatial learning strategy as a series of steps:

1. Select the material to be represented.
2. Decide the level at which to represent the text.
3. Identify at least two concepts and the relationship between them.
4. Graphically represent the concepts and the relationship identified.
5. Identify and graphically represent one or two new concepts and a relationship.
6. Repeat step 5 until all of the concepts identified in step 1 have been represented.
7. Check the graphic representation to verify that the relationships match those expressed in the text.
8. Store the graphic representation so that it can be amended when new information is acquired and retrieved for later use

Figure 2.3. the algorithmic strategy for using a graphic organizer as given by Goetz (1984)

In addition to learning the structured graphic technique, then, students are expected to apply a supporting study strategy which they also must master. The total effort expended simply to learn the strategy may be extensive. Some authors suggest semester-long courses in a strategy may be necessary before the technique can be effectively applied (Holley & Dansereau, 1984c, p.69). In fact, the importance of the graphic itself within the strategy is somewhat secondary: it is the procedure of constructing the graphic which promotes learning.[18]

This is the discipline of the formalism role: the activity of transforming knowledge into a particular form reifies and consolidates that knowledge in the mind of the student. The actual form and accompanying study strategy is something to be mastered in order to skillfully transform knowledge into the prescribed representation.

The possible benefit of the computer in this scenario is fairly obvious: by supplying features appropriate to the particular form or graphic organizer, skilled performance can be acquired much more easily. It is easier to remember or master the strategy components if a computer program will provide or prompt for them. In other words, in the formalism role, the

[18]cf. Goetz (1984) as quoted in chapter 1, p.7 of this document.

conceptual model should support this transformation of knowledge into the designated representation. Like the artifact role, the UI components should be easy to understand and use. They are something to be quickly mastered, so the student can get on with the important activity of transforming knowledge. Unlike the artifact role, the student designs the representations, arranging the concepts and ideas to be displayed.

2.6.2. Programming from formalism to medium

Programming languages are obviously a kind of formal system. Writing programs is often promoted as an intellectually valuable activity. Papert (1980) claims programming languages (LOGO in particular) provide a valuable formalism with which to interpret the world around us. Furthermore, he claims that learning this formalism is essentially learning to manipulate a symbol system, an essential part of mental development in terms of Piaget's formal operational stage.

Others have also suggested that programming provides a valuable way of thinking. In some of Papert's more recent work with LOGO, however (briefly covered in section 1.3.3), the emphasis has shifted to what the program does. That is, the original view was that creating the program was itself a good way to represent and understand things. The later view is that there is value in designing the output of a program because it allows students to construct their knowledge as they construct representations of that knowledge using computer graphics.

The former position places the software in the role of a formalism; the latter puts it in the role of medium. As a formalism, the importance of the act of programming is sacrosanct. If another, simpler mechanism for creating the same screen designs were available it could not be substituted, because the perceived value of the programming activity is that the students actually represent their knowledge in the specific form of a program. However, as a medium, the perceived value is in constructing an original representation on the computer screen, not in fitting knowledge into the formalism of a programming language. Designing the form of the screen representation requires a good deal of thought about how to portray the domain, since there is no pre-determined syntax or structure to limit how things should be represented. From this perspective, the fact that the screen design is the output of a program is not pertinent.

The next section probes the distinction between the formalism and medium roles further, clarifying the pedagogical approach corresponding to each role.

2.7. Conceptual model as medium

2.7.1. What formalisms don't do

The mastery of a specific formalism or a complex learning strategy may not the best way to promote design opportunities. Clarke (1990) claims that "deep" learning is promoted by a "struggle for form", not an automated strategy of analysis and encoding. He encourages students to consider the relative merits of different graphic organizers they could use, and even to create their own. Handing the student a ready-made formalism may deprive that student of a rich design opportunity; namely, that of reflecting on and creating the representative form.

A practical problem surrounding the use of formalisms is that teachers are unlikely to be concerned whether their students have mastered specific learning strategies to accompany a graphic formalism:

> Lindquist, Greenburg and Chickering (1983) present evidence that educational innovations are seldom adopted in toto but are rather adapted for use in new situations. The extent of diffusion depends on the ease of adaptation - for example the ability to use *part* of the innovation with an increase in effectiveness.
>
> Adoption of spatial strategies seems to me to be analogous. Identification of key elements that can be effectively adopted without the major components involved in developing skill in an entire system seems likely to be a key to the spread of schematic strategies (McKeachie, 1984, p. 303)

There could be several reasons for this; for example, a teacher usually knows what strategies s/he finds useful, and will interpret any new techniques in terms of these familiar and successful procedures. Furthermore, a teacher's perspective on the value of the graphic technique is likely to be different than a researcher's. Research in graphic organizers concentrates on the innovative learning strategies that accompany a formalism. The primary interest of the teacher in the technique may be as a vehicle to motivate and empower the students to work creatively (and productively) with minimal guidance.

These concerns can be eliminated if the conceptual model allows the user to control the form of a representation in addition to what is represented.

This is a shift from formalism to medium; stipulating a form and analyzing knowledge in terms of that form fulfills the formalism role. Advocating the design of a new form as a valuable learning exercise is the role of a medium.

Under the medium role, the conceptual model must provide the opportunity for the student to create different representations of ideas. This "struggle for form" implies that the conceptual model provides the ability to arrange interface components in different ways and to interpret them in different ways. Software that attempts this is discussed next.

2.7.2. BOXER: software as medium

Programming entails representing concepts. The LOGO programming language was specifically created as a formalism for exploring ideas in geometric and mathematical terms. Harel & Papert (1990) now promote it as a design platform with which students learn by creating instructional graphics.

The LOGO conceptual model can be used in the medium role because its components - the turtle and command language - are so general they can be applied to almost anything; that is, to creating any graphic design. However, users do not manipulate the screen objects themselves, but must go through the intermediate step of programming the turtle to produce a screen design.

The BOXER system (diSessa, 1985) is more interesting in terms of its user interface choices. The screen objects it provides are versatile and can be used in many different ways by simply manipulating the screen objects. When even more versatility is required, the additional programming capabilities are also available.

Two aspects of BOXER are particularly striking. First, it is an attempt to built a comprehensive environment for computer use in the classroom - incorporating simulations, programming, word processing, electronic mail, and so on. Because it requires a broad conceptual framework with a wide range of application, its conceptual elements are "flexible", in the sense that they can be interpreted in different ways in different contexts. For example, the basic structure is a nested container called a "BOX" which is used as a document, a file structure, a program segment, a data element, a library, a window, and so on depending on what the user is doing.

The second respect in which BOXER is illuminating is that it attempts to apply specific UI principles to achieve its objectives. The BOX structure is visually obvious (what the authors call "naive realism") in all of its incarnations. The functionality of the system is provided by a few simple operators, which can be applied generally across contexts ("de-tuning"). Rather than have a plethora of powerful operators, the same functions are available through combinations of weaker operators ("functional diffusion").

The sense of combining and re-interpreting components in different contexts is the essence of the medium role. This is the "design" element - the ability for the user to represent their own concepts or meanings with the components of the conceptual model. The BOXER effort identifies specific design principles (naive realism, de-tuning, functional diffusion) which provide a design medium and at the same time make the conceptual model easy to grasp, even for young students.

In contrast to LOGO, this software is not based on claims about the intrinsic value of programming. The value lies in the ability to reinterpret components of the conceptual model - what this author has termed *extensibility*. In BOXER, this is supported by the under-determined nature of the BOX and other constructs, which allows them to depict different concepts in different contexts.

The conclusion to this analysis is that the value of under-determined components needs to be investigated independent of the activity of programming. It is important to discover whether such interface components may contribute to constructive activity, as suggested here, without relying on the structure provided by a programming formalism.

Before pursuing this line of investigation, the four roles of the framework, which have been discussed independently, will be considered in terms of their relationship to each other.

2.8. Synthesis: the framework

Textbooks, blackboards, teachers, experiments, lockers - have different purposes. Yet computer software can fulfill some of the functions normally associated with each of these things. The role software fills is a matter of choice - of a teacher or a student or perhaps it is a matter of consensus.

The user interface is an important ingredient in how software is deployed. Figure 2.5 outlines characteristics of the user interface under each role of the proposed framework, as uncovered by the preceding review. It is hoped that eventually such characteristics could be used in a number of ways: as an aid to software developers, to help in choosing software for educational purposes, or to provide insight into new ways of exploiting software in the classroom. However, as the next section will point out, categorizing software in this manner can be complicated.

2.8.1. The selection of role

While this literature review has tended to describe software as subscribing to one particular role, it is important to keep in mind that the software does not dictate people's activity. A word processor might be used to keep accounting records, for example. A drawing program would seem to naturally fit the medium role, however, it could easily be used to draw semantic networks or other formalisms. Simulation software is the prototype of the knowledge role, but consider a meteorologist using a weather simulation. This simulation is used in the artifact role, to study a particular weather pattern, not to learn about the domain of meteorology.

These are examples of software which can be used to fill different roles. Things are even more complicated when one considers software which can be modified to do different things. Consider expert systems in the classroom. One approach is to let the students create the knowledge base, under the premise that this will help develop learning skills. This puts the expert system shell in the formalism role. Alternatively, an instructional designer or teacher could enter topical information and present the completed knowledge base as a curricular resource. The expert system's role has become that of knowledge.

Expert systems are only one example of software that can be added to or modified to suit different roles. Authoring tools and hypertext allow content to be added. Microworlds can be built using LOGO. In such cases the software serves in one role for the person who modifies, and another for the person who uses the modifications.

A further complication is variation within each category. For instance, compare the two examples of formalisms: SemNet and programming languages. SemNet has a few primitives and one basic appearance. Programming languages have a much more formidable grammar, which can produce infinite variety of behaviour and appearance for students' designs. Thus, programming languages were discussed in the context of the medium role as well.

It is clear that the roles cannot be considered to be exclusive categories of software. Instead, they are distinct ways in which software can be used (or more precisely, the conceptual model can be used) to support a pedagogical environment. Some software, like LOGO, may have potential to be used in different roles, whereas others, such as Thinkertools, may only be successful in one particular role.

This thesis does not propose that software should be developed with one specific role in mind. It may be the case that this would produce the best software, but it is not the argument advanced here. The point here is that different user interface characteristics are important depending on the role of the conceptual model. A comparison of these characteristics follows next.

2.8.2. User interface issues

Software that can be used for a particular purpose may not always provide the best features or characteristics for that particular purpose. For example, a standard drawing program could be used for the purpose of drawing semantic networks. SemNet, being specifically designed for this purpose, automatically lays out nodes and arcs on the screen, and enforces a standard labeling convention. With a drawing program, such things must be done by the user, and are fairly tedious.

In an analogous way, different UI characteristics are important for different roles. If software has an unsuitable interface for the role it is fulfilling, changes to the interface may be apropos. Because the role of the conceptual model reflects a different way of fitting software into the educational environment, each role carries a different set of priorities for the UI. In the knowledge-as-design view, each role reflects a different piece of the student's design act.

	ARTIFACT	KNOWLEDGE	FORMALISM	MEDIUM
on	*fixed in software*		*built by user*	
s	*familiar to user*	*challenging to user*	*familiar to user*	*challenging to*
l	• "metaphor" or expert model	• "phenomenon"	• symbol system	• design environment
of l	• chosen for familiarity • deliberately changes the way a user perceives task	• chosen for new learning • conveys established model of domain	• chosen for epistemic value • provides a representative form for knowledge	• chosen for design sc • provides design opportunities for the user
t	• artifact assists design act	• conceptual model is the design	• design uses given formalism	• new forms designed
	• effectively perform task	• acquire conceptual model for its own sake	• fit concepts/knowledge into the given descriptive system	• create own representa for concepts/ideas
ive	• use it • manipulate end product	• learn it • examine conceptual model	• transform into it • reify knowledge	• design in it • consider form of kn
y	usability	honesty	completeness	extensibilit
is:	• concrete association to aid abstract manipulation	• analogue of real-world object, event or abstraction	• generalized container/ placeholder for concepts	• different things in di contexts
to l	• conceptual model supports the UI objectives	• UI articulates expert model to be learned	• UI supports entry of data in appropriate form	• UI supports different its elements
es	• make task easier to understand and manipulate via conceptual model	• make conceptual model easier to understand and manipulate ("conceptual fidelity")	• create formalism which can be applied across domains	• give users choices in representative form

Figure 2.4. Characteristics of the pedagogical roles;
Differences in the conceptual model, and implications for the user interface.

Figure 2.4 synopsizes how different roles suggest different priorities for the user interface. A single suggestive term is used to capture the priorities for each role. These terms are elaborated as follows:

Artifact role: *usability* . As an artifact, the conceptual model presented by the software provides a tool for manipulating some end product. The artifact should make the user's task easier, by representing pieces of the product or operations to be performed with easy-to-understand UI elements. Any conceptual model that enhances usability is valid, since educational objectives and the student's design act are separate from the conceptual model; they must be provided by the task. Relevant questions for this role are: Does the UI make the task easier to perform? Does it help the user understand the task?

Knowledge role: *honesty*. Under this role, the conceptual model is the design which expresses knowledge to be learned; the student's task is to unravel this design. This role is much more central to the design act, and this is reflected in the user interface, whose elements represent aspects of the real-world phenomenon being modelled. The UI must convey the prescribed model as honestly as it can. The relevant questions are different in nature: Does the UI provide an accurate reflection of the conceptual model? Does it fairly represent all important aspects of the model? Does it prevent misconceptions on the part of the user?

Formalism role: *completeness*. When a student's design act is transforming knowledge into a prescribed form, the conceptual model must provide the capabilities to represent that formalism. Some screen elements must be containers or placeholders for whatever concepts the user wants to represent. This creates two concerns: the interface must adequately distinguish different concepts, and at the same time it must not exclude any idea the user might want to express. This sense of completeness is exemplified in the work of Rao & Turoff (1990), who categorize various hypertext systems according what type of ideas can be portrayed[19].

The UI issues revolve around completeness: Are all parts of the formalism accessible to the user? Is there anything that cannot be expressed? Can concepts be represented in an appropriate form?

[19]This work is reviewed in a later section.

Medium role: *extensibility.* For the user to design the screen objects, the conceptual model must not prescribe how ideas are to be represented. Some elements of the user interface must be open to interpretation. Programmers can manipulate very fine-grained elements such as lines and colours, but underdetermined UI elements (e.g. BOXER's box) should provide a better focus on the act of design.

The main issue here is whether extensibility has been achieved: Which elements can be used and combined in different ways? Do these combinations actually provoke new ideas? Will users explore these combinations?

It could be argued that each of these priorities (honesty, completeness, extensibility) is merely a different extension of usability, beyond what is reflected in conventional HCI literature and the artifact role. From this position, the point is that usability must be interpreted differently depending on which role is exercised. For example, it is difficult to see how honesty would apply to an artifact, or extensibility to a body of knowledge. The important user interface issues arise not just from the task and the conceptual model, but from the role that conceptual model plays in the act of acquiring knowledge.

2.8.3. Principles: new advice for UI developers

User interface development uses a mixture of general principles and more specific guidelines (cf. Baecker & Buxton, 1987, p. 483), which rely as much on the insight of the developer as on reliable prescriptions. The framework introduces new concerns for the development of user interfaces, which will yield additional principles. As with current practice, correct application of these principles will still require insight and innovation.

Some of these principles have already been identified in this review. It is generally clear that the four roles have different concerns, and these can be addressed by the interface. The principle for developers would read *the role of the conceptual model should be supported by the user-interface*. The HCI literature has also produced some suggestions on how to support each role. There is no shortage of information on how to make an artifact more usable. Less frequently, the notion of conceptual fidelity appears in the literature, which is reflected by the notion of "honesty" in the framework.

The idea of formalisms and making them complete has to be inferred from a variety of existing systems and associated writings; this is the first time issues surrounding this kind of software have been presented collectively. The medium role goes even further; extensibility is an original synthesis of software capabilities that have proven important. The idea that it can be supported by the user interface is barely hinted in previous constructs such as "functional diffusion" and "de-tuning" (DiSessa, 1985).

The framework also suggests that pieces or elements of the user interface can be used to provide the overall qualities of usability, honesty, completeness or extensibility that are desirable. Norman (1983) asserts that the conceptual model should correspond to the observable parts of the target system. These parts of the target system subsequently appear as pieces of the user interface. Although conceived in terms of the artifact paradigm, this analysis can apply to other roles as well. Under the framework of figure 2.4, elements of the UI correspond to the spirit of each role. They can be pieces of a task (artifact), pieces of a phenomenon or model (knowledge), placeholders for concepts (formalism) or building blocks for new representations (medium). Clearly, this has a crucial effect on how these elements should appear and behave. The fact that UI components need to be interpreted in terms of the conceptual model's role can be asserted as a principle: *The role of the conceptual model should be manifest in UI elements.*

The ideas of under-determined and undetermined features are the new contributions of this work. The remainder of the framework's suggestions are a systemization of ideas that are already extant in the literature and existing software. Thus, the principle that remains to be tested is: *under-determined features provide the user with the opportunity to design new representations.* The provision for the user to introduce new designs for ideas has been stressed throughout as the defining characteristic of the medium role. Whether underdetermined features actually help generate a constructive "struggle for form" in the student cannot be assessed from any material that has been reviewed. This requires an original investigation.

2.8.4. Furthering the framework

The framework is not intended to be exhaustive in enumerating different types of software. Nor is it the final statement on how computers can be used in education. Rather, it highlights certain issues and raises questions that are obscured by current practices in user interface research. Some of these ideas, like conceptual fidelity, have been explored in independent projects, or identified under the auspices of more popular concerns such as "user-friendliness". Others, like extensibility and under-determinism, appear to be original with this work.

The framework can be extended in several ways. An exhaustive review of existing educational software would clarify the prerogatives of each of the four roles. There is a need to tie the roles to pedagogical styles and specific curricular units, to provide some practical benefits. Before these pursuits, however, the radical claims of the framework should be substantiated. This thesis takes the latter direction, in search of undetermined features and extensibility.

2.9. Hypertext as experimental platform

Any type of software could be used to explore undetermined features. For instance, undetermined colours or sounds could be added to a word processor, and their use investigated. There are two reasons why hypertext was the ideal choice for experimenting with undetermined features. First, the basic hypertext constructs of node and link provide natural candidates for undetermined features. While often implemented in a well defined or carefully structured manner, these basic entities are themselves very general. This reasoning is demonstrated in section 2.9.3. The second reason for choosing hypertext is that its potential as a constructive environment has been widely recognized, making it a good candidate for the medium role.

Kozma (1991) talks about the potential of hypertext which allows "the learner to add their own information and construct their own relationships, perhaps symbolically representing them with node and link structures." Note the similarity between this vision of hypertext and the research in graphic organizers. Kozma goes on to claim: "Such systems can be made to correspond to the processes learners use when constructing interrelationships among concepts in real memory." (p.203). This same faith underlies the belief that graphic organizers model actual semantic memory structures. Even the "link" and "node" terminology (Quillian, 1968) is essentially the same in hypertext and network memory models.

The knowledge-as-design outlook does not require such a strong correspondence to justify the value of hypertext. It takes after the enactive approach of Soloman (1988): that actually manipulating physical analogues of concepts will help learners to think about them, irrespective of whether or not the system is an accurate model of semantic memory processes. That hypertext structures do have an impact on students' structuring of domain knowledge has been empirically established (Teshiba & Chignell, 1988; Gray, 1990).

Developers of hypertext systems usually adhere to the holy grail of usability.[20] Nevertheless, other concerns are recognized by those who discuss the student as hypertext author. The framework allows us to examine the user interface in light of these concerns, considering hypertext first in the role of formalism, and then as medium.

2.9.1. Hypertext as formalism

One of the appeals of the formalism role is that it provides structure with which to create designs. Without a specific design space, the design task may seem too vaguely defined to be of practical use. In terms of hypertext, this formalism perspective suggests suitable structures should be enforced by the software. Such is the position of Rao and Turoff (1990) : "Hypertext should be treated as a general purpose tool with approaches to handling nodes, links, and retrieval, that fits within the context of any application and conveys common meanings to the users. To accomplish this we need a comprehensive framework for Hypertext based on a cognitive model that allows for the representation of the complete range of human intellectual abilities." (p.335)

Using Guilford's structure-of-intellect theory, Rao and Turoff proceed to enumerate the factors which comprise human intelligence. Hypertext systems are evaluated according to how well they cover this list of factors.

Their main analysis consists of classifying hypertext components according to the list of factors. Hypertext primitives (nodes and links) are associated with the *operation* dimension of Guildford's theory. Different types of primitives are required to span the *product* dimension for each

[20]e.g. from Nielsen (1991), p 190: "from the discussion of empirical usability research...there is no clear evidence for whether hypertext is in fact superior to paper." Note the terms usability and superiority are almost treated as synonyms.

primitive. Generic primitive types are not considered sufficient: "We contend that single node types are inadequate because they do not allow representation of the complete range of cognitive abilities" (p. 342)

This thinking clearly corresponds to the formalism role. Like Kozma (1991), these authors believe the explicit structure of hypertext corresponds in some fashion to human memory. If this structure is incomplete in some way, then the formalism is deficient: "most current systems do not span the range of cognitive processing abilities that humans are capable of... While it is common to get around this problem by adopting ambiguous design metaphors (e.g. node, box, link), we feel that this produces a number of problems...it is extremely important that the meta-functionality of cognitive relationships among objects be fully and explicitly defined within the system." (p.352).

It is difficult to translate Rao & Turoff's analysis into implications for the user interface. Their criterion for existence of a cognitive factor within a hypertext system appears to be very weak (or possibly undefined). For example, in some of the systems they analyze, different link types are distinguished by no more than a label attached to the link. Yet this is considered sufficient to encompass different factors of the model.

It would seem to be a simple matter for a user to attach an appropriate label to a hypertext link, should the system in question have such under-determined components. It is difficult to accept that merely hard-coding these labels into the software will miraculously endow the capability for representing intellectual abilities.

Formal hypertext structures, such as that suggested by Rao & Turoff, provide a specific categorization for links and nodes. Just as this formalism defines specific constructs for authoring hypertext, it curtails some design possibilities of hypertext as a medium. This is not to say providing such structure is a mistake, only that it is not suitable for the medium role. Systems that are suited to the medium role are considered next.

2.9.2. Hypertext as medium

Authoring is restricted to what Nielsen (1991) calls hypertext *engines*. These engines allow any number of documents to be entered into the same basic software. Hypertext systems which are designed for specific documents and contain linking mechanisms particular to one context do not allow the user to create their own hypertext designs[21].

A second distinction is between plain systems and alterable systems. Plain systems provide for different documents, but they always use the same mechanisms and interaction techniques. For example, a link in the Guide system always produces a pop-up window in the top right corner of the screen (Nielsen, 1991, p.111). Links always are activated in the same way, and always behave in the same manner. Alterable systems are essentially programmable[22]. The programming language allows the behaviour of links and various screen objects to be modified.

Alterable/programmable systems clearly provide more design possibilities than plain systems. However, even programmable systems tend to restrict the "look and feel" of the final document to a hypertext model provided by the system. HyperCard, for example, is built around the metaphor of a stack of index cards. By definition, activation of a HyperCard link brings up a new card. Buttons, text fields, graphic drawings, and menus are the screen objects which are chosen and placed on each "card" by the hypertext author. The document itself fills in the content of the objects (i.e. the text in text fields, the images in graphic drawings).

To change how an object behaves, the author resorts to programming. This means the author does not deal with the components of their design, but with the fields and buttons of the hypertext systems in combination with other basic constructs of the programming language. In addition to the extraneous effort of programming, this interferes with the interpretation the author wishes to make of the screen objects. In HyperCard, for example, one could program turned-down corners to mark index cards of interest. The fact that the turned-down corner is "actually" an altered HyperCard button may be transparent to a casual user of the document. However, *the author who created this design* spends their time programming it as a button, rather than thinking about the implications of their design.

---

[21] e.g. Palenque, Drexel Disk (Neilsen 1991, p.112)

[22] HyperTalk is the programming language of HyperCard, and InterLisp in the case of NoteCards. HyperTalk and HyperCard are registered trademarks of Apple Computer, Inc. InterLisp and NoteCards are products of Xerox Corporation.

This reinforces the idea of using under-determined features instead of programming to provide extensibility. In other words, there are two possible ways to provide extensibility with hypertext:

- *Behaviour*. Alterable hypertext systems allow you to change the behaviour of the screen elements, which may then represent new concepts. However, the complexity of such alteration interferes with simple association of concepts.
- *Appearance* Hypertext systems (either alterable or plain) could have under-determined screen elements. The student changes the appearance of these components to suggest different meanings.

Obviously, there are plain systems which do not support the medium role. Guide, for example, provides three types of links, each for a different purpose. This repertoire restricts the author to the three basic link meanings provided. On the other hand, Brown University's Intermedia system defines an integrated linking protocol between programs. This allows the same basic link mechanism to serve in different contexts and be used in different kinds of structures. These different contexts are provided by a multiplicity of separate editing environments and applications.

With plain hypertext systems, untyped link and node constructs are required to provide the full design capabilities of a medium. Of course, there is no guarantee of extensibility - authors may still have problems instantiating new concepts creatively with links and nodes. The following section provides some speculative examples of designs that could be produced in hypertext with undetermined features, without relying on programming or on multiple environments.

2.9.3. Designing without programming

The following examples (figures 2.5.1 through 2.5.3) are intended to illustrate how plain hypertext can support a conceptual model in the medium role. Under-determined positioning of links, undetermined link symbols, and unspecified node content allow the author to arrange these elements according to their own design. Meanings for these features are decided by the user in the context of their use.

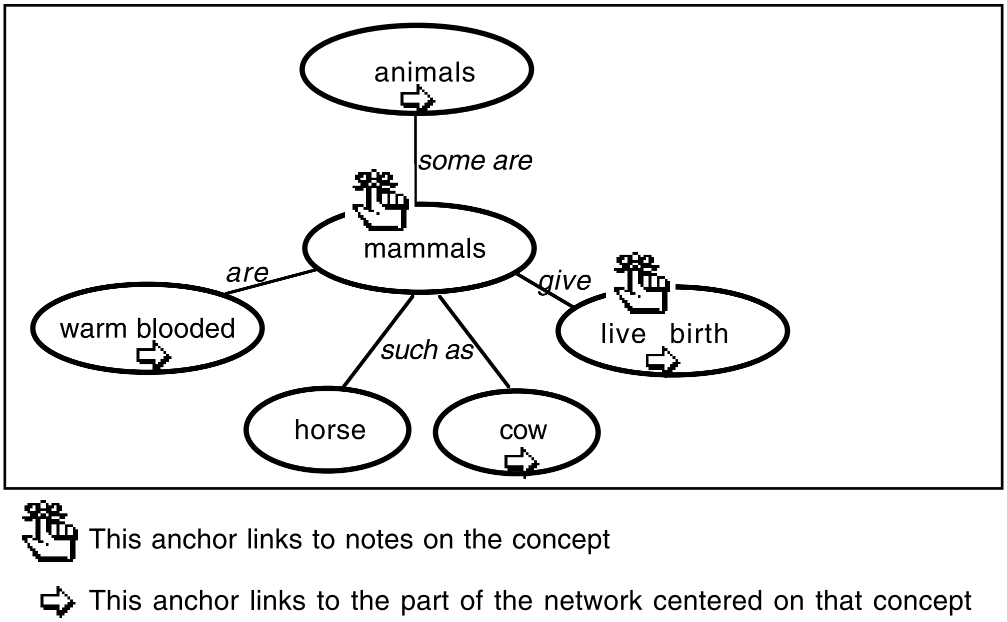


Figure 2.5.1 A semantic network built in hypertext

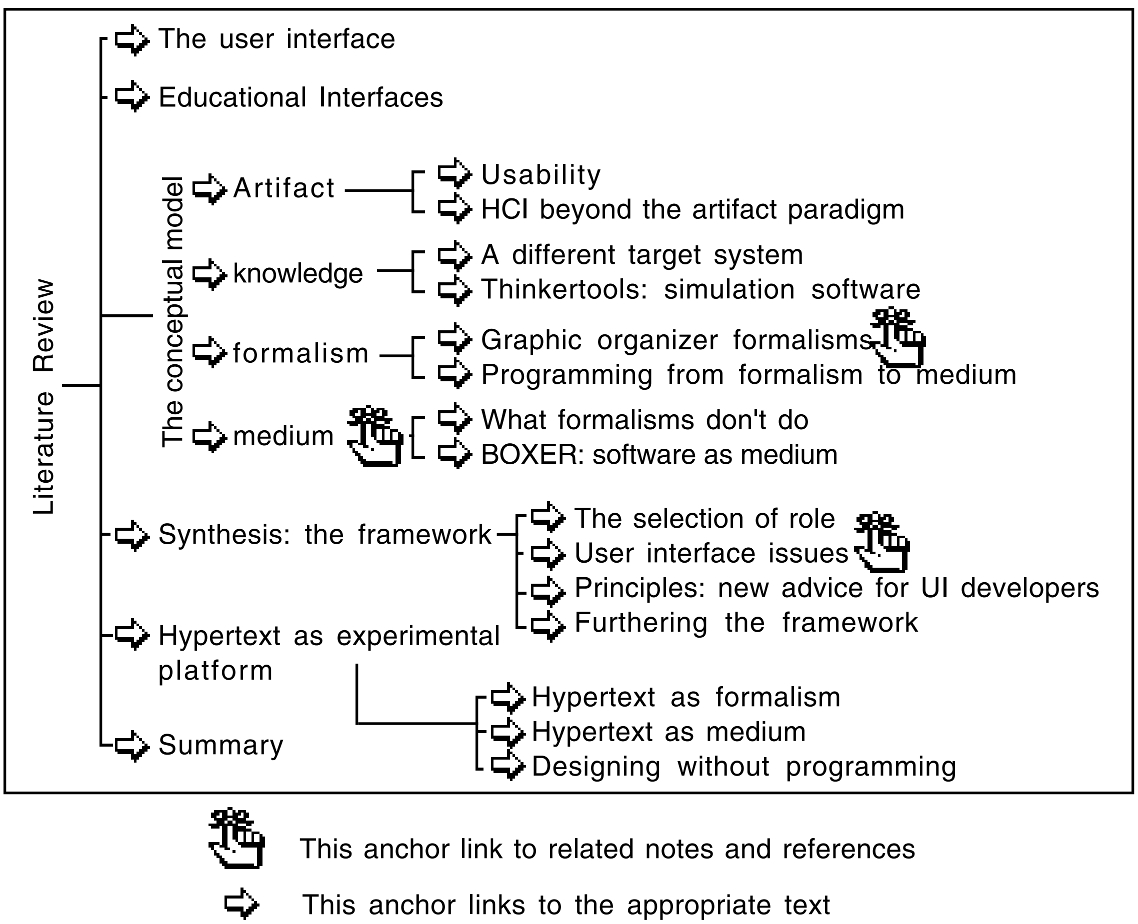


Figure 2.5.2. Table of contents in hypertext.

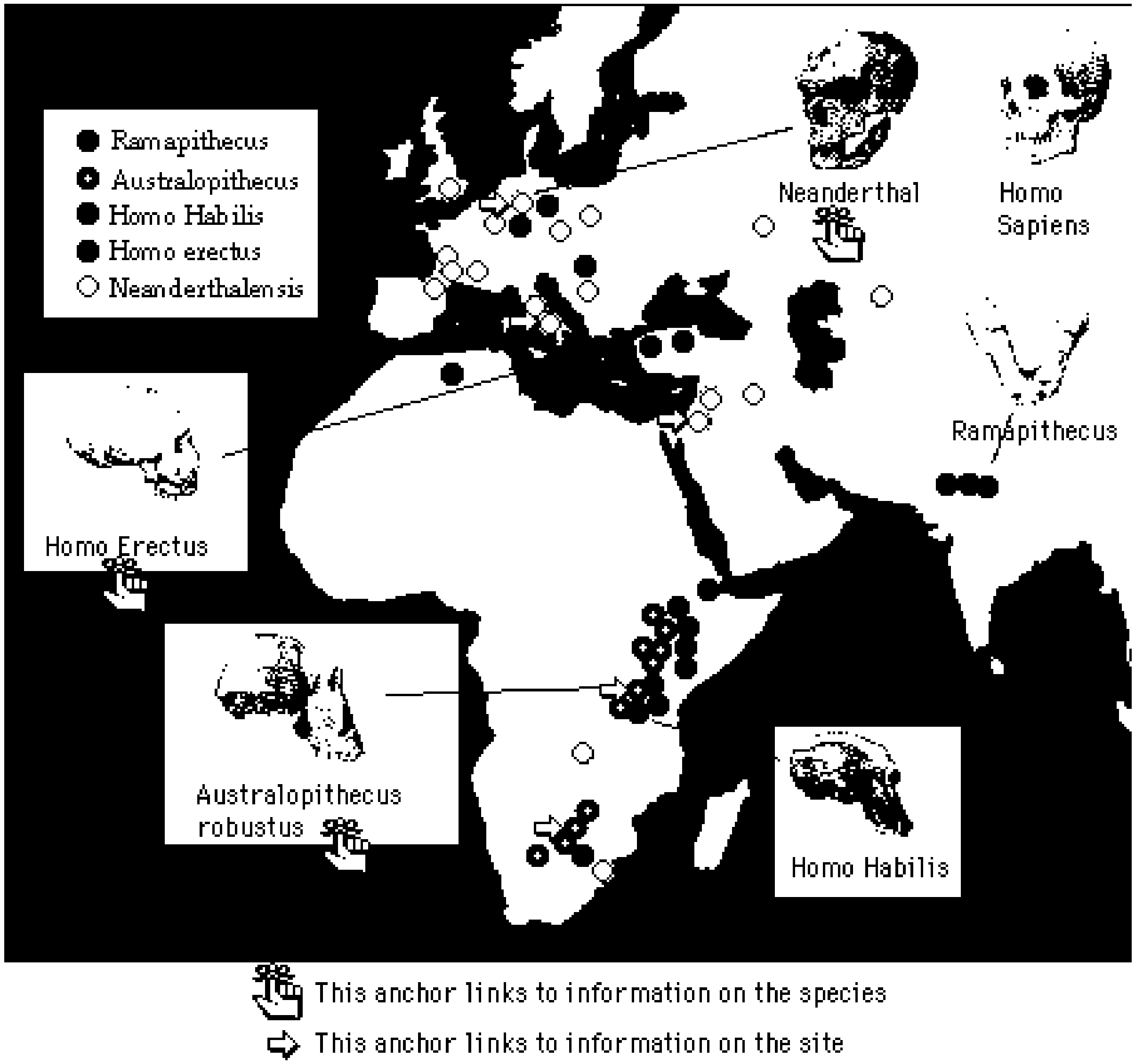


Figure 2.5.3. Hypertext links added to a conventional map.

Each example provides a different context for the interpretation of links. These designs have the flavor of graphic organizers; the appearance and structure of the graphic adds meaning to the elements making up the graphic. However, in each case, the structure is determined not by a formalism enforced by the software, but by the author of the hypertext document. The positioning and meaning of links, as well the type of information contained by a node are choices made by the user. The ability of the user to create different forms and meanings by combining simple screen elements makes the medium role tenable.

The first example shows the use of undetermined features in hypertext to create a semantic network. This allows more elaborate networks than a planar paper representation, since new nodes will provide the equivalent of an infinitely extendible "piece of paper". Nodes are used to show a central concept and the connections to that concept. One anchor symbol is used to navigate to another concept, and another anchor symbol is used to provide detailed notes on a particular concept.

The second example shows a table of contents, with links from the section headings into the document text. A different link symbol is used to access background and references for certain topics. This organization and the semantics of the links are entirely different from the first example, simply by using the same node and link mechanisms to represent things in a different way.

In the third example, the node uses a graphic form that already has a conventional interpretation - a map. In the two previous examples anchor symbols distinguished different kinds of navigation. In this case, anchor symbols are used to distinguish between textual elaboration on different topics (i.e. sites vs. species).

Clearly paper-and-pencil forms can be mimicked on a computer screen; the capability to reproduce graphic organizers is inevitable. The real issue is how software capabilities and restrictions can provide additional design value. The examples above are intended to demonstrate how hypertext primitives can be exploited without restricting the hypertext author to one particular form or organization of knowledge. Instead, the context and the author's design provide semantics for links and nodes.

Specifically tailored mechanisms for linking are not necessary. Special software capabilities are not needed. All that is needed to encompass such variety is undetermined features which are open to different interpretations; that is, which do not dictate the structure of the design to its author.

2.10. Summary

This review has examined how software can contribute to different parts of the student's design effort, with the conceptual model of that software fulfilling different roles. The observation that the user interface directly contributes to the success of software under each of these roles leads to a framework for identifying important aspects of the user interface, depending on the role it fulfills.

Some guidelines for user interfaces in general, and specific characteristics for user interface components are suggested by this framework. Of these, the proposition that underdetermined components can produce extensibility in the interface remains unsubstantiated. To pursue this idea, an experimental platform is needed.

Widespread interest in hypertext authoring as a constructive learning activity leads to its endorsement as a test platform for extensibility. Particular elements of hypertext systems - nodes and links - are natural candidates for undetermined features. The specific implementation of these features is discussed in chapter three.

3. Design of the Experiments

3.1. The Environment

A new hypertext environment was created for this experiment. There are three traits in particular that were difficult to find in existing hypertext systems.

First, there had to be provision for undetermined features. Most hypertext engines provide a strong model of nodes and links. For example, a node may be modelled as a "document" or a "note". This prevents users from attaching their own interpretation to objects. Systems without a dominant model of hypertext are often large and complex (e.g. Intermedia).

The second concern was to avoid the formalism style of interaction, in which users are simply translating knowledge into a known structure, rather than thinking about the design of that structure. This can happen even when some underdetermined feature is available. NoteCards, for example, allows the user to assign link semantics in the form of text labels. Unfortunately, this labeling tends to be used in an obvious manner, as a means of grouping note-cards. In contrast, the new environment uses undetermined anchor icons, in the manner demonstrated in section 2.9.3. The choice of icons is contrived to be an integral part of link authoring. The user is almost forced to consider their interpretation, simply in the course of operating the interface.

The final concern was a constraint of the experimental design; namely, the system is intended to be used almost immediately, without learning a large number of features or operators. Otherwise, the users are subjected to the same problem as programmability; they have a lot of skills to acquire before they can begin designing. The experiments use no operations or features that are not represented on the screen or manipulable by direct exploration via point-and-click mouse operations. It provides what diSessa (1985) calls "naive realism". This also means that the use of this system is not exhaustive, in the sense that many possibly complex features (e.g. full text search) are not used; the experiments confine themselves to the "naive" features of the hypertext system.

3.1.1. Undetermined Anchors

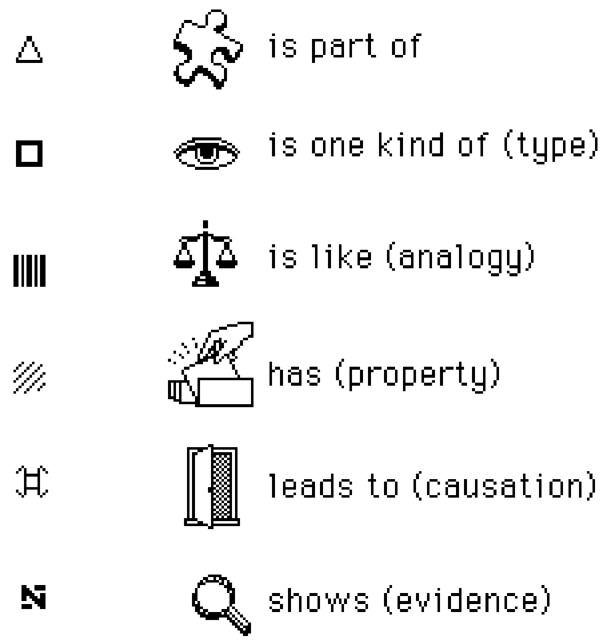


Figure 3.1. Different anchor appearance can be used to provide undeterminism or a pre-determined set of link types.

The examples of figure 2.5 have shown that anchor symbols can be used in dramatically different ways. Although it may be useful to select symbols with some mnemonic value, such as those to the right in figure 3.1, there is some danger that symbols may bias subjects' designs, by suggesting particular meanings. To avoid this problem, ambiguous symbols were used, like those to the left in figure 3.1. It was reasoned that this presented the best chance of observing genuinely original designs from subjects.

Even without particularly mnemonic symbols, there are a number of reasons such anchor symbols are an appealing choice for undetermined features:

- The basic system functionality is unaffected by the use of different symbols. As essentially an add-on feature, it is relatively simple to isolate its effect from the other features of hypertext.
- The hypertext product is essentially the same, so results may have implications for the use of hypertext systems in general.

- As a simple variation on hypertext, it demonstrates the elegance of using the UI to provide extensibility, rather than introducing complex software mechanisms.

These arguments could easily be applied to textual link labels. However, people seem to make a special effort to understand what symbols represent. Text or number labels simply do not seem as incongruous when they appear on the screen. This may also apply to the mnemonic anchor symbols, in which case there may be some additional constructive advantage to using the ambiguous ones.

Extensibility is provided by the appearance of links. The behaviour of the links is discussed next.

3.1.2. Choice of Linking Mechanisms

The linking mechanism must be sufficiently general so that it can reflect different semantics without changing how the links actually behave. That is, the behaviour of links must be interpretable according to whatever anchor symbol the user chooses and whatever meaning the user assigns to that symbol. Furthermore, the desire to have a "zero learning" interface means the mechanism must be simple and obvious.

In most hypertext engines, links connect a single anchor point to a hypertext node. In the program built for this thesis, link connections are similar to those in the Intermedia system, and are made between two anchor points. This provides several advantages over the anchor-to-node method:

- the behaviour of the link is visually obvious (naive realism)
- an anchor on the destination node can locate a specific item of interest
- "back-ups" to previously visited nodes are available with no additional mechanism (functional diffusion)
- the anchor-to-node style of hypertext can be emulated in this situation, whereas anchor-to-anchor cannot be emulated in the reverse situation.
- the behaviour of the link is the same at both ends.

Most importantly for this research, both ends of a link are visually conspicuous as anchor symbols on their respective nodes, making the undetermined feature particularly salient.

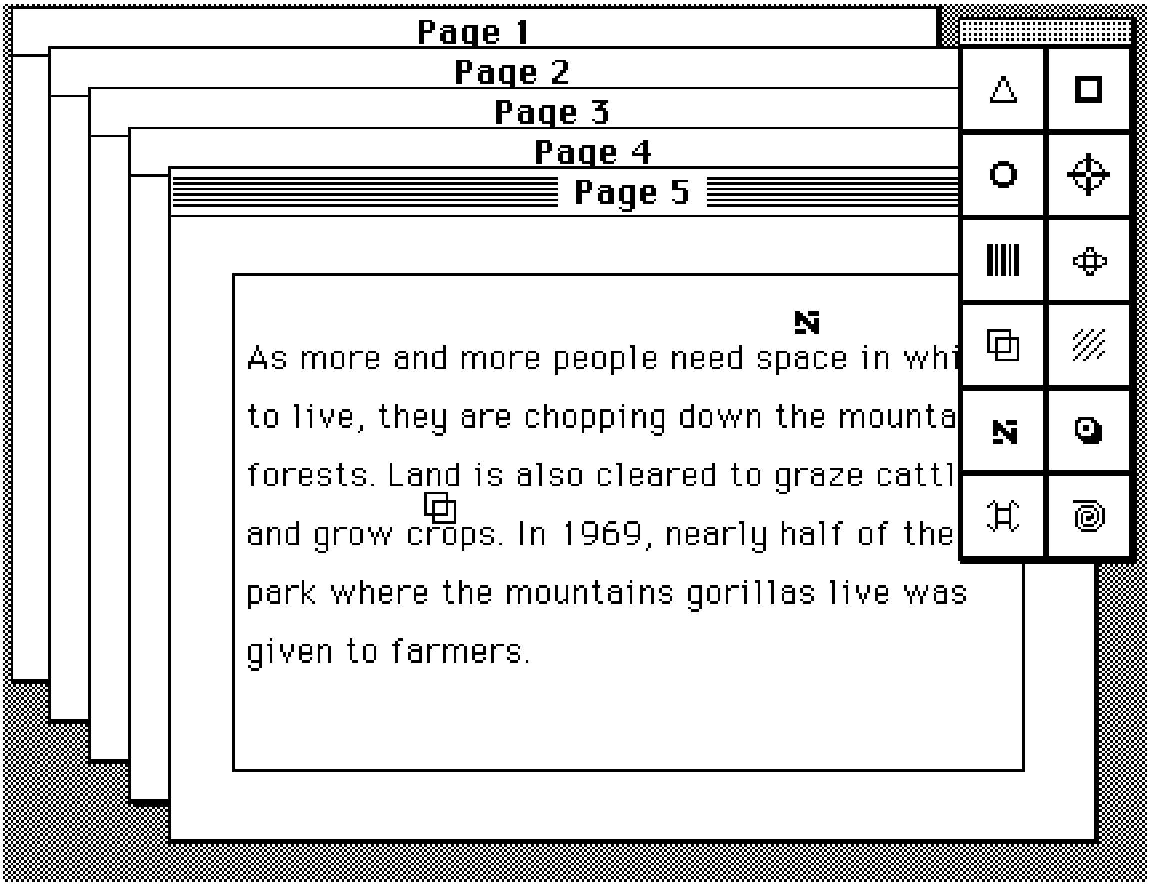


Figure 3.2. A typical screen from the hypertext system used in this experiment. A palette of link anchors appears to the right.

The interface is designed to be trivial to operate, in order to accommodate naive users and experienced computer users at the same level of competence. To construct a link, one selects from the palette of anchors (shown to the right in figure 3.2) and then clicks the mouse at the two desired locations for the link anchors. To operate a link, one simply clicks at an anchor and the location of the other anchor will appear. Sound and highlighting provide feedback regarding what operation the user is performing, and help the user locate anchors and nodes.

Even the basic functionality of these few simple operations provides a rich design space. This discussion now moves to the specific task which circumscribes the design possibilities, making it practical to study specific features of the interface.

3.2. The Task

An experimental task was needed in which the subjects actually used undetermined features. The main problem in choosing this task was that the undetermined features do not work in isolation - different anchor designs are suggested by different node designs, which relate to different content, and so on. It was important to establish conditions under which outcomes could be attributed to some undetermined feature, and not to intervening factors, such as the greater design space created by using features in combination, or to the novelty of using a flexible environment.

For this reason, the experiments focused exclusively on the undetermined anchor symbols. The structure and content of the nodes were restricted to ordinary text, without any accompanying graphics. This makes the experiment much more controllable; subjects did not alter the hypertext content or the node designs - they merely selected and added links. While this did not provide the constructive freedom that section 2.9.3 speculates about, it did help isolate a specific contrast for study.

This contrast is between the use of undetermined anchor symbols and the use of anchor symbols with predetermined meaning. A bonus of using this contrast is that the software was essentially unchanged in all experimental conditions. This was accomplished by simply assigning meanings to the symbols for the predetermined (or control) condition. Under the treatment condition, subjects were required to interpret the anchor icons for themselves, rather than using prescribed interpretations. Thus, the experiment eliminated any concern that outcomes might be due to differences in the way different software packages operate.

The task instructions made sure the subjects dealt with the knowledge content of the hypertext document, and not simply with the mechanical action of linking. Two kinds of evidence were sought that the treatment actually made a difference; recall of the document content and verbal reports during the task performance were collected in separate experiments.

The recall data provide insight into the students' acquisition of knowledge. While it is hoped that subjects' recall will be better under the treatment of undetermined features, it seems probable that the extra demand of creating meanings may interfere with the time the subject has to study the document contents. Therefore, the subjects' retention of the content is used as a more likely indicator of the benefits of the treatment. The time-limited controlled-contrast experiments were not adequate to test other learning outcomes, such as transfer or synthetic reasoning.

Of additional interest is demonstrating that a difference in the design activity actually occurs, which could account for any difference in recall. While this can be inferred from the constructs used in the design of the experiment, verbal reports reflect on the integrity of those constructs and on the theory behind them. That is, the verbal reports track how design activity was altered by the treatment.

The remainder of this chapter details the design of these two experiments.

3.3. Two experiments

Two experiments were used to examine the effect of assigning meanings as an additional requirement in the creation of hypertext links. In both experiments, subjects were divided into two groups. One group assigned meanings to the link symbols as they used them, and the other group used meanings which were predetermined for each link symbol.

In experiment A, the ability of subjects to recall the content of the hypertext document is considered. For this experiment, a post-test control-group design is used, producing quantitative results which are assessed with simple statistical tools.

The second experiment probed the task process directly rather than examining recall outcomes. Evidence in the form of verbal protocols was gathered concerning subjects' thinking during the link creation task. These qualitative data were then analyzed in terms of the theorized effect of assigning meaning to hypertext links on the way in which subjects think about the content of the hypertext document.

3.4. Experiment A

3.4.1. Hypothesis

The main hypothesis is that creating meanings for anchor symbols will produce better retention of the hypertext material than using predetermined meanings for anchor symbols.

3.4.2. Subjects

Thirty-two adult subjects participated in the study, drawn from the student populations of two eastern Canadian universities. Fifteen subjects were members of a graduate course in education, and the remainder were respondents to an advertisement.

All subjects had previous experience using computers, although the range of experience was broad. Some subjects had been professional programmers, while others were familiar only with word processing. None of the subjects had extensive exposure to hypertext applications.

Subjects were randomly divided into control and treatment groups. Subjects in the control group were to use pre-assigned meanings in the linking task, and those in the treatment group were to design their own meanings for the symbols they used.

3.4.3. Procedure

All subjects performed two separate tasks: a linking task and a reading task. The experimental conditions were manipulated in the linking task, whereas the reading task, which involved no linking whatsoever, served as a baseline for evaluating recall performance.

Each subject's recall performance under the linking task was considered relative to their recall performance without the linking component of the task. Alternatively, a standard comprehension test could have been used to gauge subjects' normal study performance. However, the procedure used offers a couple of advantages; first, there are fewer possible explanatory variables introduced, since the same software and environment are used for the baseline measurement and the experimental conditions; second, the data produced by the reading and linking tasks can be compared directly, since they are in the same form. As a bonus, it is possible to compare each subject's performance on the linking task to their "normal" study behaviour.

Having two separate tasks introduced a number of complications to the experiment design. Since each subject had two independent tasks, two hypertext documents had to be used. The order of the tasks, and the document used for each task introduced two additional independent variables. Along with the condition variable, this yielded a 2 by 2 by 2 experimental treatment design.

There were a number of steps for each subject in addition to those already mentioned. Before the actual experimental tasks, each subject was trained in the use of the software, and introduced to the concepts of nodes, links and anchors as instanciated in the program. As well, each subject participated in four separate recall tests; immediately following each of the two tasks, and a repetition of both recall tests one week later. The delayed tests provided data concerning subjects' retention of what they had remembered.

Each step in the procedure and their sequence will now be explained in some detail. The script used to administer the experiment is given in appendix B.

Step 1: Habituation to the software environment

This step familiarized the subject with the basic operation of the hypertext program, as well as the basic concepts of links within a hypertext document. Without a general idea of what links might be used for, subjects are not able to perform the linking task in any meaningful way. The idea of a link was introduced as a connection between ideas in the text.

Subjects were provided a hypertext document, complete with links. After a brief introduction and demonstration of the program, the subject was asked to find answers to specific questions by following the hypertext links. The questions appeared on the screen in sequence. When the subject visited a node believed to contain the answer to the question, s/he then proceeded to the next question. The number of correct responses and a countdown timer were displayed on the screen. During this period, the experimenter interacted freely with the subjects, answering questions and correcting any problems the subjects had in operating or understanding the program. Ten minutes was allocated for this stage.

Following this hands-on task, there was a short review of the concepts of links, and the idea of using different anchor symbols for different meaning was discussed.

Step 2: The linking task

Before beginning the actual linking task, subjects were shown how to make links within the program. The investigator made a number of links as a demonstration. Subjects then practiced making links for a total of five minutes.

The instructions for the linking task asked the subjects to "connect ideas on different pages together so people could find ideas by using the anchors. This could be similar to how you used the anchors when you were answering questions in the training task." Regarding the use of link symbols, subjects were instructed to "use one anchor symbol for each linking meaning."

The handling of conditions was next. This involved a variation in instruction between the control and treatment groups. The treatment group was given a sheet of link symbols to refer to when making their links and told "Many people find it convenient to write down the meanings they use for each symbol. Here is a sheet on which you can do this, if you like. " The control group had meanings already printed on the sheet, as shown in figure 3.3. They are told "I will be giving you linking ideas to use."

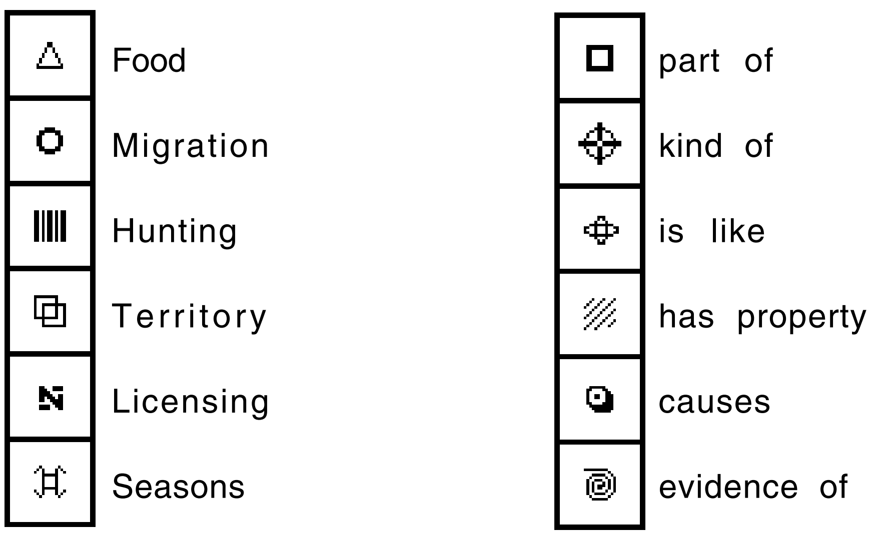


Figure 3.3. Symbols and meanings used by subjects in link creation. For subjects linking the alternate set of materials, the six meanings available in the left column were Extinction, Food, Poaching, Habitat & Location, Protection, Numbers & Statistics.

The link meanings that were given to the control group are the six most popular among the meanings generated from the treatment group, accounting for 85 per cent of the links made by treatment subjects.[23] The

[23]This method of selecting meanings required that the treatment group be completely finished before any control subjects began the experiment.

remaining six meanings are those advocated by Holley & Dansereau (1984b) as being "close to optimal" for semantic networks.

Subjects were given 15 minutes to perform the linking task. Before they started, a preview of the material was provided. Each hypertext node appeared on the screen for 10 seconds. Once the last node was seen, the subjects began their linking task. This preview was intended to give the users a sense of the size of the document, and to reduce the feeling of disorientation often associated with reading hypertext documents. (cf. Gray, 1990)

Step 3: immediate recall of link task content

Upon completing the linking task, subjects were asked to recall as many ideas from the text in as much detail as they could. These recollections were recorded by the subjects in longhand, on standard writing paper. No time limit was imposed for this step.

Step 4: study task

A comparison study task was performed by each subject. The intent was to reproduce the conditions of the linking task, but without the linking requirement of the task. Subjects were simply told to study the text, and that this would be followed by a recall test.

A second document, with no links, was provided for this task. As with the linking task, a preview of the material was provided. Once again, subjects were given 15 minutes to work with the material. They used this time to visit the nodes and review the content as they liked. Linking was not possible during this task.

Step 5: immediate recall of study task content

The recall instructions of step 3 were repeated following the study task.

End of session one

Steps 1 through 5 were performed in one session lasting approximately one hour and forty minutes, varying slightly depending on the time the subject required to complete the recall steps. For half the subjects, steps 4 and 5 were performed before 2 and 3, and the other half completed the steps in the order given above.

Session two was held one week after the subject attended session one. Recall was again recorded in the same manner, adding steps 6 and 7 to the procedure. These steps were ordered according to the task order in session one. Session two typically ran thirty minutes.

Step 6: delayed recall of linked task content

The recall test was repeated. To insure the order of recall, the subject was specifically instructed to recall the linked material at this time.

Step 7: delayed recall of study task content

Using the same instructions, the subject was asked to recall the unlinked material.

While keeping to the prepared script, the experimenter interacted openly with the subjects in response to questions, and in response to any difficulties during the training portions of the experiment. Of course, during the actual tasks and recall tests there was no communication with the subjects.

3.4.4. Analysis Methods

Independent Variables

There were three independent variables in this experiment. The *condition* variable determined whether the subjects created their own meanings for link symbols. The *order* variable determined whether the linking task or the study task was performed first. The *material* variable indicates which of two sets of material was used for each of the two tasks.

Each of the three variables has two treatment levels, giving a 2 by 2 by 2 fully crossed design for a total of eight cells. Four subjects were placed in each cell. Subjects were assigned to levels of order and material at the time they performed the experiment, randomly placed into a cell which remained to be filled (i.e. had less than four subjects). For the condition variable, treatment cells were filled first, and then control cells.

Dependent measures

The dependent measures were based primarily on the subjects' recall performance. In order to obtain a measure of this performance, each recall composition was examined in order to identify specific ideas from the original text. This method required two separate stages, according to the following description.

First, the original text was segmented into "idea units" as suggested by Lloyd & Steele (1986). To generate text segments for experiment A, the text of the hypertext documents was printed on paper. The instructions for segmenting this text were to "draw a line to separate 'idea units' within the text. An idea unit is a separate idea, concept, or fact in the text which would count as something additional to remember." Replication of this segmentation produced 92 per cent agreement between six independent workers (on average). The segmentation actually used for analysis purposes is given in appendix C.

The second stage was to identify these idea units in the recall compositions. Each recall composition was transcribed to computer disk for analysis. Where a composition contained an idea from the source text, a code corresponding to that idea unit was inserted into the recall at that point in the composition. Separate codes were used for idea units when they were only partially recalled by the subject, and also when the recall of the idea was faulty. To determine reliability of this coding method, an independent rater coded 20 per cent of the compositions, resulting in 87 per cent agreement between the two raters.[24]

Dependent variables are based on accurately recalled idea units as indicated by this recall coding (that is, codes for partial and faulty recall were excluded). The total number of ideas recalled by each subject yields four variables, one for each recall test. Represented as proportions of the total number of ideas in the text, these variables are called *link immediate, link delayed, study immediate* and *study delayed*. The main hypothesis deals with subjects' retention of their recall across from one week to the next. The variables *link retention* and *study retention* express the number of ideas recalled in the delayed test as a proportion of the number recalled in the corresponding immediate test. The difference between these scores, (link retention – study retention), gives the improvement in retention of the

---

[24]Note on coders: The six coders who identified idea units were drawn from the general university population, as were the subjects themselves. Coding of recall was performed by this investigator, and the second independent coder (used for reliability) was a psychology major uninvolved in this research. To eliminate bias, coding was performed blind; that is, during the coding process coders had no idea to which subject group each transcript belonged.

linking task over the study task, which was called *link improvement*. The distribution of link improvement between control and treatment conditions allows the main hypothesis to be tested.

Additional biographical data was gathered on all subjects, and the computer program produced timing information of all subjects' activity in the event these data should prove useful. Data concerning the number and types of links used, and the amount of time spent and number of visits to each node was automatically collected in this way. A stopwatch was used to time the recall tests, although no time limit was imposed. Subjects in the treatment group were asked to provide meanings for any symbols they had used for links, if they had not previously recorded them. Finally, informal written feedback and interviews concerning the experiment itself were also collected from some of the subjects after the experiment was complete.

3.4.5. Materials

Two sets of materials were created, drawn from sources at the eighth grade reading level. The topics chosen were "the mountain gorilla" and "the polar bear". Each topic was divided into ten pages, or nodes, for the hypertext document, as shown in appendix C. Subsequent segmentation of the text apportioned 57 idea units to the first set and 55 idea units to the second set, suggesting amount of factual content in the two sets is roughly the same.

3.5. Experiment B

3.5.1. Purpose

The first experiment was designed to collect quantitative evidence of improvement in remembering due to the treatment condition, that is, resulting from subjects creating their own meanings for the links. In this experiment, subjects' cognitive activity is examined more directly. Through the use of verbal protocols, evidence is gathered concerning the effect of the experimental conditions on the subjects' cognitive process. Thus the question addressed is not whether the treatment affects recall performance, as in experiment A, but whether subjects' thinking about the task is altered in the manner theorized.

3.5.2. Subjects

Six graduate student volunteers were used in this experiment. All were proficient in the use of the computer and pointing devices, and all were familiar with the idea of hypertext. Two subjects were computer science students, with an extensive programming background, and the remaining four were education students. However, none of the subjects had any extended experience using hypertext systems. Three subjects were placed in each of the control and treatment groups.

3.5.3. Procedure

The experimental procedure was much the same as experiment A, with some changes: the recall tests and study task were eliminated. Consequently, no second session was required, and only the first set of hypertext material was used. Instead of these tests, subjects followed a "think-aloud" verbal protocol to report their thought processes during the linking task. A post-task, unstructured interview was also conducted in which subjects reconstructed the strategy they used in making links.

The task was extended to thirty minutes in anticipation of extra time needed for verbalizations. Videotape was used to record the verbalizations and activity of the subjects during the entire session, which took approximately fifty minutes.

The steps as outlined below maintain consistent numbering with steps of experiment A. The complete script is provided in appendix D.

Step 1A: *Habituation to the software environment*

The introduction to the program and habituation task were exactly as in experiment A.

Step 1B: *Training on verbal protocols.*

This part of the procedure was absent in experiment A. Subjects were told that the investigator was interested in what they were thinking during this task, and they were told to "Think and reason aloud, and report everything that passes through your mind while you are working."

The investigator then modelled this task, making a number of links while thinking aloud. Subjects were then asked to practice the protocol and task using an alternative set of material. During this practice, the investigator interacted freely with the subject, offering suggestions and encouraging them to verbalize their thinking. When both subject and investigator were satisfied with the subject's use of the protocol, the experiment proceeded to the next step.

Step 2: *The linking task.*

This step proceeded as in experiment A, with an additional fifteen minutes, for a total of thirty available for the task. The only change to the task instructions was to remind subjects to "think out loud" during the task.

Unlike experiment A, the investigator was permitted to intervene to remind the subject to continue verbalization. The phrases "keep talking" and "tell me more" were typical of the prompts used when the subject lapsed into silence for more than ten seconds.

As in experiment A, control subjects used predetermined link meanings, and treatment subjects invented their own meanings for the link symbols.

Step 3: *Interview.*

A post experiment interview of an informal nature was used to attempt to establish global strategies the subject used in the task. It was anticipated that the verbal reports would contain primarily information concerning the subjects' immediate attention, and therefore some additional data would be useful in any attempt to reconstruct more global intentions. In this interview, subjects were encouraged to recall their strategy for link construction in a step by step manner, in terms of what they did first, what they did next, and so on. The interview typically lasted 10 to 12 minutes.

3.5.4. Analysis Methods

Transcription and segmentation of the verbal data was straightforward. A transcript was used to aid in the segmentation procedure while viewing the videotape. Segmentation instructions were to "divide the verbal transcript at each new thought. Pay attention to grammatical cues, such as conjunctives which combine more than one idea and verbs which indicate a separate sentence. Pauses and reflective utterances such as 'um' or 'ah' are also good indications that the subject has moved onto a new thought". A second rater repeated this procedure on one-third of the verbal data. (That is, one-third of the data as measured by the first rater's segmentation.)

There was 94 per cent agreement between the segmentation of the two raters. [25]

Further analysis of these verbal segments required derivation of a coding scheme. When a well defined problem or task is being performed, it is possible to trace the subject's progress through various partial solutions to the problem. Verbal segments can then be coded according to the subject's transition from one partial solution to the next. Such schemes are not appropriate here, since the task is amenable to different interpretations, and therefore the work cannot be encoded consistently between different subjects.

A coding scheme suggested by Ericsson & Simon (1984) identifies different types of information being verbalized, without reference to the specific components of the task. Adopting this scheme distinguished four categories of "verbalization" segments:

- *Intention* describes an action the subject will take, or a state to be achieved. Verbalizations such as "let's take this.." qualify for this category.
- *Attention* indicates the subject is examining some aspect of the current situation. Simply reading or reviewing an item would qualify.
- *Evaluation* compares the quality of alternatives or indicates a decision of some kind, and may be indicated by simple explicatives such as "yes" or "OK".
- *Planning* involves exploring the effect of different actions mentally. "if I do this then.." or "how can I get.." are examples.

These categories do not help in analyzing what source of information subjects heeded; or alternatively, what components of the environment they incorporated into completing the task. Since this is relevant to the purpose of the experiment, the coding scheme was extended. A cursory examination of the protocols suggested that there were a relatively few

---

[25]Note on coders: Because of familiarity with the video tapes, this investigator was unable to effectively blind his coding in experiment B. Therefore, segmentation and coding was performed by a psychology major unfamiliar with the experimental hypothesis or expected outcomes. This investigator's segmentation and coding was only used for reliability comparisons.

categories needed to cover the elements discussed by the subjects within the segments.

The "component" categories used are:

- *Content* when subjects read or discussed what the text actually conveyed.
- *Link* when the segment refers to an existing link or one that could be created.
- *Type* of link is in evidence when the segment refers to the symbol or meaning selected for linking.
- *Network* when a subject discusses the pattern of links in the document as a whole.
- *Operation* is used for simple discussion of how the program works, rather than the linking task.
- *Meta-comment* was used to code segments about the experiment itself, or unrelated external issues.

It was found that under the link and network categories, segments often involved relationships within the text, without specifically mentioning the link mechanism. Rather than include the inference that these segments are actually about making links, and to avoid making that inference part of the coding scheme, two additional categories were adopted:

- *Relationship* is used when two parts of the content are compared, but no specific mention of the link mechanism is made.
- *Global* is used when the segment refers to the text as a whole without reference to link mechanism. The pattern of ideas or facts is dealt with, without the pattern of links as context. Segments which refer to one part of the text, such as a single node, are coded *content*.

Each segment was coded on both dimensions; the verbalization type (intention, attention, evaluation, planning) and component mentioned (content, link, type, network, operation, meta-comment, relationship, global). An uncoded category was also included for each dimension.

Segmented transcripts and the videotapes were used in coding the data. The video proved useful to disambiguate references in the protocols, such as "I'm selecting this now." The coding scheme was presented and discussed with the coders, and a number of segments were used to model the coding process. Sample segments for the coding categories are given in appendix G.

All the material was coded by one individual, and one-third of this material was recoded independently, achieving an overall agreement of 84 per cent between the two coders.

3.5.5. Materials

The first set of materials from experiment A was used in experiment B.

3.6. Summary

This chapter has explained the development of a hypertext system and two experimental designs suitable for examining undetermined features. The hypertext system incorporates characteristics particularly relevant to the experiments, encouraging constructive activity within a rich design space while keeping the requisite computer skills to a bare minimum.

By constraining this study to the examination of a single specific component (i.e. the anchor symbols), it was possible to construct direct contrasts between conditions which either do provide or do not provide undetermined anchor symbols. One experiment investigates subjects' recall for a learning (or more specifically, memory) effect, and the other looks at the effect of the treatment on process; on how the subjects actually perform the task. Chapter four reports the results of these experiments.

4. Results and Discussion

This chapter presents the results of the experiments and the subsequent analyses which reflect on the theoretical claims under investigation. Including automatic data capture and informal observations, the mass of data which could be presented is quite formidable. The data used in this presentation are either outcomes which are explicitly part of the experimental design and are therefore essential for interpreting the results, or they are supplemental and elaborate on the claims of the thesis. Thus, data for which this investigator was unable to find implications relevant to this work are not reported, allowing a relatively concise report.

The findings for experiment A are presented first, followed by those for experiment B. Elementary interpretation of these results help consolidate their presentation, while broader implications and conclusions are left to chapter five.

4.1. Experiment A Results

Figure 4.1 shows the recall score variables introduced on page 56 - *link delayed, link immediate*, *study delayed* and *study immediate* . These graphs plot delayed against -immediate recall performance, and provide a comparison of the two tasks[26]. While the difference between the two tasks is striking, it is the range of the data values that is somewhat disconcerting. A low performance might be attributed to subjects' spending more time on the creation of links, and somewhat less reflecting on the content of the text.

However, this is hardly a suitable explanation for the two lowest scores (indicated as outliers in figure 4.1). These subjects recalled only one or two ideas from the text. Such extreme low cases may indicate some fundamental difficulty in performing the task - that is, subjects did not understand the instructions, and simply made links in a mechanical manner, without regard to content .

These two subjects with both made comments, following the recall test, which suggested such fundamental problems. One remarked that he had not "understood the purpose of it" (referring to the linking task) and the second subject stated that he hadn't expected the recall test and he would

[26]- These values are based on the raw scores given in appendix F

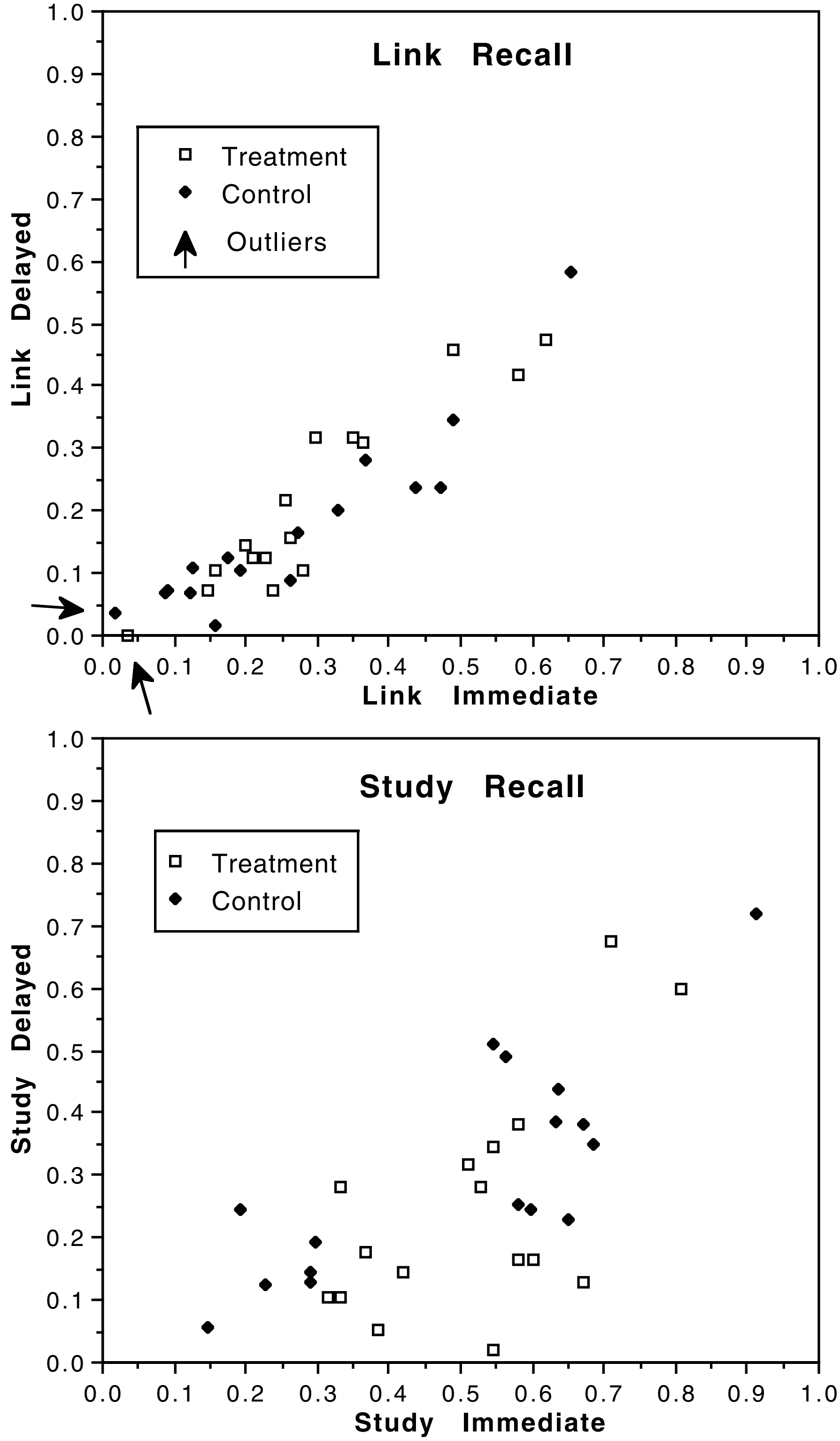


Figure 4.1. Proportional recall scores for link and study tasks, immediate versus delayed. The horizontal axes represent the proportion of total ideas that were recalled by the subject in the immediate recall test. The vertical axes represent the proportion of total ideas that were recalled by the subject in the delayed recall test. The proportion is found by dividing the number of ideas recalled by the total number of ideas in the original text.

"do better next time", incorrectly assuming that the linking task would be repeated in the second session. Although these two subjects were not native English speaking, other subjects who made similar comments or were not native English speaking did not produce such low scores, so it is likely that these two subjects represented extremes in performance or comprehension of the task.

In figure 4.2, the distribution of the dependent variable *link improvement* shows these two observations as extreme values within their respective conditions. Comparison of these values with the interquartile range of their entire group identifies them as possible outliers on this variable (cf. Tukey, 1977)[27]. Combined with the low performance scores and suspicious subjective evidence, this strongly suggested removing these data points from the data set.

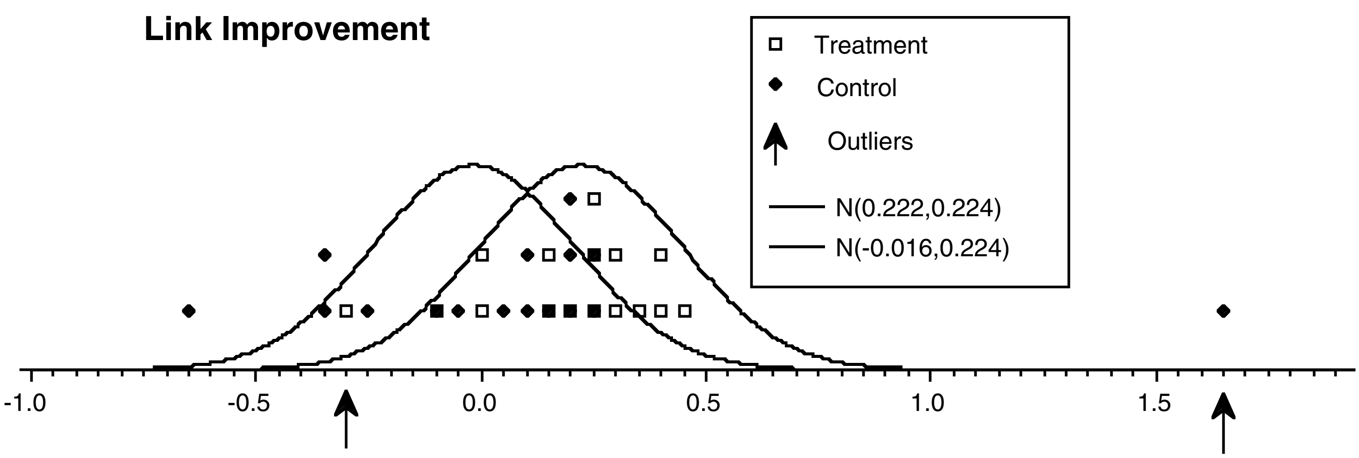


Figure 4.2. Dot plots for the distribution of Link Improvement variable: control versus treatment groups. Observations have been rounded to the nearest 0.05 to produce a rough display of the distribution of observed points. Filled squares occur were the distributions of the two groups overlap. The normal curves reflect means and pooled variance for the two groups with the two outliers dropped.

Removing the two outliers reduced variance substantially, and the null hypothesis (which is based on *link improvement*) could be rejected (p <

[27]Data points more than 150% beyond the interquartile range (that is, the middle 50% of the data points) are considered possible outliers according to Tukey's method. Only the two data points indicated in figure 4.2 are identified as possible outliers by this method.

0.007) instead of accepted (p > 0.41), according to a simple t-test. In the remainder of this discussion, these outliers have been dropped from the statistical tests.

A clear interpretation of the link improvement variable requires some consideration of its constituent variables, *link retention* and *study retention*. These two variables can be contrived as the ratio of the delayed to immediate scores which are depicted in figure 4.1. Analysis of variance results for all three variables are shown in figure 4.3.[28]

		Link Retention			Study Retention			Link Improvement		
Source	*df*	*Adj SS*	*F*	*P*	*Adj SS*	*F*	*P*	*Adj SS*	*F*	*P*
condition	1	0.0311	0.68	0.419	0.2372	3.90	0.061	0.4403	8.91	**0.007**
order	1	0.0827	1.80	0.193	0.3534	5.82	**0.025**	0.0941	1.90	0.181
material	1	0.0159	0.35	0.562	0.0131	0.22	0.647	0.0001	0.00	0.959
condition x order	1	0.0182	0.40	0.535	0.1123	1.85	0.188	0.0401	0.81	0.378
condition x material	1	0.0560	1.22	0.281	0.0250	0.41	0.527	0.1557	3.15	0.090
order x material	1	0.0254	0.55	0.465	0.0012	0.02	0.888	0.0377	0.76	0.392
condition x order x material	1	0.0020	0.05	0.834	0.0008	0.01	0.907	0.0056	0.11	0.740
Error	22	1.0106			1.3370			1.0878		
Total	29	1.2565			2.0626			1.8255		

Figure 4.3. Abridged analysis of variance results for three principal dependent variables - those reflecting on the main experimental hypothesis. Significant effects (p < 0.05) are indicated in bold typeface.

The results for the link improvement variable support the experimental hypothesis; subjects in the treatment group (i.e. those that created their own link semantics) improved their retention performance over the study task more than subjects in the control group (i.e. those using assigned link

[28]With the outliers dropped, the ANOVA design is unbalanced, with two cells having 3 observations and the remaining six cells having 4 observations each. Thus, a standard ANOVA procedure cannot be used, and the sum-of-squares values must be adjusted for the unbalanced design. These results were produced using the general linear model (GLM) procedure of the Minitab software package.

semantics), with $p < 0.01$. None of the other effects reach statistical significance for this variable.

The results for the link retention variable are less supportive. On average, the treatment group does outperform the control group in retention under the linking task (see figure 4.4), but this difference does not reach statistical significance. This is not really damaging to the study, as the design anticipated that there would be a wide range of performance due to individual characteristics. This was the reason for the link improvement construct and the introduction of the study task as part of the experimental design. Each subject's recall performance under the linking task is measured relative to their performance in a neutral study task, via the link improvement variable.

Source	*level*	Link Retention	Study Retention	Link Improvement
condition	control	0.621	0.637	**-0.016**
	treatment	0.691	0.469	**0.222**
order	link first	0.712	**0.658**	0.054
	study first	0.600	**0.448**	0.152
total		0.656	0.553	0.103

Figure 4.4. Means for dependent variables across levels of condition and order. Significant differences ($p<0.05$ from figure 4.3) are indicated in bold typeface.

The F value for the condition under the study retention variable, although not quite significant, is large enough to be uncomfortable. Since the two groups were undifferentiated for the study task, there should be no effect of the treatment group on this outcome. In fact, this is the only independent variable which was deliberately undifferentiated for the study task. The main concern is that the two groups were unintentionally distinguished in some manner by the experimental procedure or instruments.

The only difference between the two groups intended by the experimental design is that all treatment subjects were run before the control subjects. This was necessary to collate the link meanings to be used in the control condition. It is possible that the investigator's instructions or presentation varied in some manner as more and more cases were performed. However, the order of the subjects does not correlate highly with their study retention score, so this does not appear to be a likely explanation.

The remaining possibility is that some factor external to the experiment was introduced because assignment of subjects to conditions was not sufficiently randomized. A number of possible explanatory factors, including age of the subjects, their educational background and the fact they were drawn from two different universities, were examined in an attempt to discover a source for this difference, but without success.

The hypothesis suggests that had the treatment group performed as well as the control group on the study task, then they would have performed still that much better on the link task. That is, if the means had been equal for the study task, the improvement due to the linking would show up in the link retention variable. As it stands, the numbers indicate that link improvement in the treatment group not only compensates for the lower scores in the study task, but additional increase as well. This can be seen more clearly in the figure 4.5.

Source	*df*	*Adj SS*	*F*	*P*
Study Retention covariate	1	0.3740	11.00	**0.003**
condition	1	0.1399	4.12	0.055
order	1	0.0000	0.00	0.971
material	1	0.0051	0.15	0.701
condition x order	1	0.0004	0.01	0.906
condition x material	1	0.0948	2.79	0.110
order x material	1	0.0308	0.91	0.352
condition x order x material	1	0.0035	0.10	0.751
Error	21	0.7138		
Total	29	1.8255		

Figure 4.5. Analysis of variance for link improvement, with study retention as a co-variate.

The analysis of the variance of link improvement in figure 4.5 includes the study retention variable as a co-variate. Although most of the contribution of the condition variable has been subsumed by the study retention variable, the F value for condition is still quite high. Thus, the effect of the condition is not due only to the lower scores on the study retention variable.

Of the remaining independent variables, only the order of the two tasks produces a significant effect (with $p<0.05$) in figure 4.3. Study retention is significantly better among those who have the link task first. This can be explained simply as a practice effect. Subjects that have previous experience with the recall test have a better idea what they are preparing for in the study task. Thus, they can do a better job of preparing for the test. It is interesting to note that this effect does not appear in the link retention variable. This can be attributed to the difference in the task itself; for the study task, the subjects have no substantive purpose other than to prepare for the recall test; in the linking task, subjects have to focus on a separate, substantial problem. Thus, familiarity with the test procedure does not affect the way a subject approaches the linking task, as s/he has something to do other than preparing for that test.

Under this interpretation, it should be possible to reduce the order effect by familiarizing subjects with the recall test before the first task begins. However, this effect does not appear to substantially influence the overall results.

4.1.1. Caution on interpreting the results

It is important to remember that the significant improvement demonstrated by these results cannot be interpreted simply as "improved recall". The measures examined in the preceding figures deal with proportion of an initial recall that can be reproduced after one week. While it is true that the treatment improves this proportion, care must be taken in drawing further inferences.

For example, the actual number of idea units recalled is much better for the study task than the linking task. This is not surprising, as the linking task is more complex and the objectives given in the task instructions are independent of recall performance. Furthermore, subjects were not given time to take the linking task to completion, whereas many subjects completed the study task before the 15 minutes were finished[29].

[29]Two subjects completed the linking task before the time expired, whereas all but one subject were satisfied with the amount of time provided for the study task. This was concluded from informal observation of the subjects' overt behaviour. Many subjects simply stopped paying attention to the task or reported that they were finished.

Following the experiment, many subjects questioned the legitimacy of comparing the linking and study tasks directly, incorrectly assuming that this was the experimental contrast. Rather, the comparison of interest pertains to the linking task; between a treatment group who created link meanings versus a control group for whom link meanings were provided. The study task is not intended as a treatment contrast, but was designed to establish a "baseline" recall performance for each subject.

In this experiment, there is no presumption that the link task will be a more efficient means for memorizing facts than ordinary study. In fact, it is fairly certain that the more constructive activity of making links and designing semantics for them will be more time consuming. The purported advantages lie elsewhere - in getting the student involved in using and integrating knowledge in a more substantial and more permanent way. In this experiment, scoring factual recall was used as the evaluative tool.

The four dependent variables representing recall scores - link immediate, link delayed, study immediate, study delayed (represented in figure 4.1) are not used to test our hypothesis, but are still of secondary interest. Their means are given in figure 4.7.

	Link Recall		Study Recall	
Source	Immed	Delayed	Immed	Delayed
Control	0.282	0.180	0.526	0.322
Treatment	0.312	0.227	0.518	0.255
Total	0.297	0.203	0.522	0.288

Figure 4.7. Means for variables representing the proportion of facts recalled.

Quite clearly, the linking activities detract from recall performance. It would be convenient to suggest that, with a long enough delay, the link task would eventually surpass the study task on this score. A simple model of memory might indeed predict such exponential decay. Alternatively, the constructivist view would advocate that recall under the link task should be superior if that task were taken to completion. These are both arguments that the link task is better not only for retention but also for recall. However, there is no direct evidence available from this study to directly support either assertion.

Some subjects do not even perform better in retention on the link task. These appear as negative scores in the link improvement variable. In figure 4.2, these observations can be observed to the left of the zero mark. In fact, the mean for the control group on this variable is essentially zero. The overall positive value in link improvement is due to the contribution of the treatment group. For the control group, then, not only is recall performance worse, but there is no overall improvement in retention. In terms of the idea that constructive activity will produce better retention, it would seem that the control group does not evince such a result. Either the subjects do not produce the beneficial constructive activity theorized, or the measurements are not sensitive to it. For the treatment group, however, there is a striking improvement in retention with about the same loss in recall performance as the control group. This suggests the treatment provides as good an exposure to the material, while improving the constructive quality of the task.

But there is no direct evidence that "constructive quality of the task" can be credited with the success of the treatment, since only achievement and not process has been examined in experiment A. Experiment B identifies this "constructive quality" in the task process - detecting what is actually different in the task activity.

4.2. Experiment B Results

4.2.1. Description of verbal data

A large proportion (40 per cent by segment coding) of the verbal reports were simple inspection of the hypertext document; subjects were either reading the text or thinking about what it said. This is an essential part of the task, but reveals little about the subject's pattern of activity .

For the design of links, subjects reconstructed three basic strategies in post-task interviews. One strategy was to select important ideas from the text, and then search for a related idea with which to form a link. The second strategy was simply to scan the text for any idea that re-occurred. The final strategy was to search for ideas based on a particular linking relationship or link meaning.

These strategies were employed differently by different subjects; one reconstructed only the second strategy, most reported switching back and forth between the different strategies, and two subjects claimed to be using more than one strategy concurrently. Under casual inspection, the think-aloud data does seem to support subjects' reconstructions. However, there is no apparent effect of the treatment on the choice of strategies; they seem to be more a matter of personal style or inspiration.

In the treatment condition, it is necessary to create link meanings in the course of performing these linking strategies. This situation is exemplified by the following excerpt from the verbal transcripts:

> Page one for example, talks about how there are so few of them. And that might be linked with the page that says how.. about the fact that there are no mountain gorillas in zoos at all. OK, I'm not sure what kind of natural link, what kind of link makes sense here. Am, that's one possibility. .. Also the fact obviously the fact that they need a very big range and they take time to forage and digest their food, the fact that's a high mortality rate, also should be connected, they are also reasons as to why they're becoming extinct. Um...Trying to see what kind of link I would do here because there are different concepts here.

This extra challenge of creating the link meanings is the way in which the treatment condition attempts to improve this constructive linking task. Contrast this with the next excerpt in which the subject does not have to pause and consider the link meanings at all:

> Page six talk about Rwanda, Zaire and Uganda, and those are specific locations, and we have no links there yet. So perhaps we should make a location link backwards from there, so that if people want to know where specifically gorillas are found they can look there. Now we're back to page two we have gorillas in the wild are found only in Africa. Let's try completing the location link.....to the word Africa, oh I see on page two we do actually introduce the actual countries so we'll start them there instead.. So now when we go from page two we want additional information on any of those three countries that were named. We can go and read a section on page six about the anti-poaching patrols. Now we want to make sure that people that are interested in poaching can get to page six and then on to page seven and/or page seven directly from the first time that idea occurs, perhaps on page one or two. Now looking at page one again, page two talks about the rarest species and where they live, page three introduces the idea of poaching. They're both slaughtered deliberately and they're killed accidentally in traps that poachers set for antelope. Let's try a link between poachers set for antelope and.. the poaching that we saw on page six, although they refer to different kinds of poaching.

These two transcript fragments demonstrate how creating link meanings can impact the linking activity. This does not mean there is a definite pattern which can be described. Just as subjects use different linking strategies, they approach the creation of these meanings in different ways. Several meanings may be generated to fit a particular linking scheme, or a single meaning may be thought up as the link is created. Furthermore, a single subject may stop agonizing over these meanings once four or five satisfactory ones have been derived.

The real issue is not to describe link creation precisely, but rather to determine whether this treatment changes how the subject deals with the knowledge content of the document. In fact, it is quite possible to struggle with the structure of the document without dealing with the link meanings at all, as in this excerpt:

> So, page two, we've got talking about them being in the wild, page ten we have this again, gorillas in the wild, so I've used gorilla families and natural movement. Why don't I just change one of these because they're too specific in a sense.. let's get rid of natural movement, I don't like that any more, let's change, I prefer that word to sort of gorillas in the wild, so I'm going to connect this one to page two. Right. And now let's check page one. The forest, with so few individuals left, now was there another one on page nine, yeah. This was also gorillas in the wild. The problem here of course is that there's too many, the combinations become too great. There three or four pages, then there's four pages, then there's gonna be something like six different links I got to make. So..I can't really do them all. I wish one square would take in all the rest or something.. gorillas group.. I'm not sure how to resolve that problem.. The problem is there are links between page one, like I've seen links between page two and nine and they should all be interlinked, it's like they should have sort of a common link between them all rather than just going from A to B.. which is kind of restrictive.. So

The coding scheme was intended to capture differences in how a subject dealt with the document, and to quantify these differences in a reliable way. Struggling with the design and planning how to link things up, for example, was differentiated from thinking about specific ideas or even about individual links. The most direct way to compare these different kinds of verbal segments is to simply count them, as reported next.

4.2.2. Frequency analysis of segment codes

Proportional occurrence of each coding category is shown in figure 4.8. Exact frequency counts for each subject are given in appendix H. Certain categories dominate by frequency, which is not surprising. As already noted, subjects spend a large proportion of their time reading aloud or talking about the content, which is reflected in the large proportion of "attention" on the verbalization dimension and "content" on the component dimension.

Comparative frequency of these categories cannot be taken as a direct indication of their contribution to the task. A single planning segment, for example, might be related to a whole series of subsequent attention and intention segments. Also, subjects vary according to how much verbalization they do, although training does help flatten this effect.

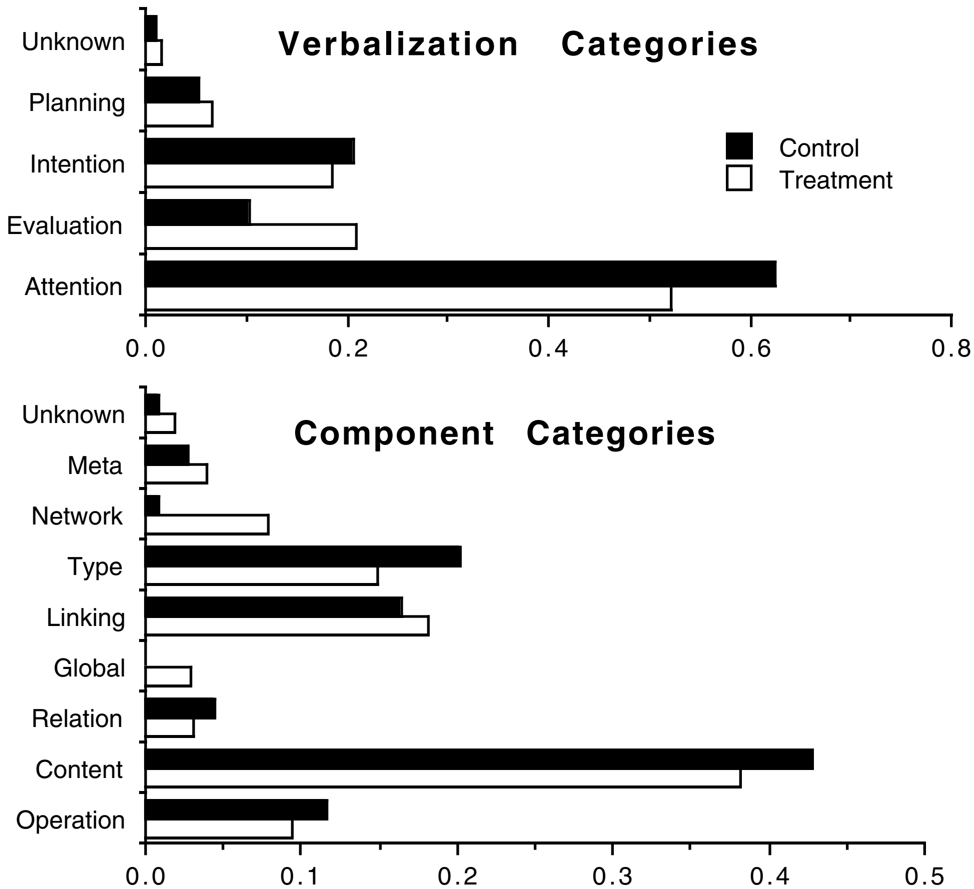


Figure 4.8 Frequency of each coding category as a proportion of verbal segments, comparing average of three control against average of three treatment subjects.

However, the conditions can be compared within each category, to reveal something about the task process. The distribution of verbal segments across verbalization type categories (attention, evaluation, intention, planning) are very similar for control and treatment groups. Only the change in proportion of evaluation segments is appreciable, with approximately double the frequency. The differences according to the other coding dimension are more dramatic.

On the component dimension there is a marked difference between the two groups on the frequency of network segments and global segments, while all the other categories remain fairly even. No global segments appear at all in the control group, and very few of the network category. Recall that the network segments deal with the pattern of links throughout the text, and that global segments also deal with ideas or themes in the text, but without reference to the pattern of links.

These values indicate that the treatment does not change the relative amount of time spent planning or evaluating information as much as it shifts these planning and attending activities away from the details and specific task items to dealing with the material as a whole.

The absence of any global segments among the control group is highly suggestive. Treatment subjects verbalize about the material as a whole, independent of reference to the pattern of links in the document, but control subjects only verbalize about the whole material in the context of links. The need to create link semantics, then, forces the treatment subjects to deal with the integration of the entire material through the use of these symbols. The control group, given link semantics, does not have to consider what the ideas are that would link the material as a whole. The ideas they must use are pre-determined.

4.3. Summary

One experiment shows an effect of the treatment on performance results; the other shows an effect on task process. As examined, both the software environment and the task display characteristics typically attributed to constructive activity. Subjects learn fewer facts in a short amount of time than in a simple study task (as measured by recall performance), but they retain the knowledge better.

The treatment under investigation improves the "constructive quality" or "design opportunity" in the task by having subjects create their own semantics for graphic elements, rather than rely on predetermined ones. The better performance this elicits in experiment A is paralleled by a change in task process in experiment B. Subjects spend more time considering the material as a whole, rather than specific parts of the material or separate components of the task. There is also a less pronounced increase in the amount of evaluation which subjects do. Further conclusions are drawn in the closing chapter.

5. Conclusions and Future Directions

The experiments used in this thesis serve to demonstrate the application of some original ideas in user interface development, which are summarized by the framework of pedagogical roles. These ideas do not constitute testable theory, but are akin to the "guidelines" and "principles" that are prevalent in the field of human-computer interaction. As such, these ideas are not falsifiable.

However, these ideas, and this framework, are based upon a constructivist approach to learning, which makes a number of claims about the way understanding and learning is best achieved. If the software correctly implements these claims, then the experimental outcomes should be consistent with constructivist theory.

The conclusions in this chapter reflect separately on each of the experiments, the specific software platform used, and finally on the motivating ideas behind all of these.

5.1. Experiment A

Improved retention of factual information was a result of the use of undetermined anchor types. As with any experiment, generalizing this result means considering the conditions and outcomes to which it could reasonably apply.

There were subjects who did not benefit from the linking task, and presumably found linking to be an impediment to learning the information. These appear to the left of the zero mark in figure 4.2. Apparently, these subjects have effective study strategies which are disturbed or disrupted by the linking task. An attempt to isolate these strategies would be useful, to compare them with the "global" thinking that the treatment imposes (as identified in experiment B). Strategies of those who benefit from linking could also be compared with those who do not benefit. Initial efforts could use verbal protocols in the fashion of experiment B.

The linking task has a general damping effect on the recall distribution, which is apparent in figure 4.1. The variation in study recall is obviously greater than that in the linking task. It seems that the linking task primarily benefits those with poor recall performance. Demonstrating that poor studiers can be selectively helped would be a powerful result, and worth the investment of further experiments. Once it is possible to discriminate those individuals who could benefit from particular constructive tasks, practical application of these results cannot be far off.

5.1.1. Value of short term studies

The size of the effect shown in experiment A - about twenty percent improvement- does not compare well to methods or mnemonic devices targeted specifically at recall. This is hardly surprising given the time constraints and the fact that the treatment is not striving for memorization.

Outcomes other than recall of facts may seem pertinent, such as evaluation, synthesis, transfer, and so on. Recall has served to identify an improvement of one kind, but the big advantages claimed for constructive approaches deal with improved understanding, not improved accumulation of facts.

An elementary measure was chosen because of the difficulty of producing measurable results in the short term using constructivist methods. While the restrictions on the time and task applied in this thesis are antithetical to the constructivist approach, they provided a controlled experimental design. With an effect clearly established, it is now possible to let subjects take their task to completion, without being concerned that the results are due only to increased time-on-task.

In order to establish "deep learning" (to borrow a term from Novak & Gowin, 1984), constructivists often adopt unconventional measures or rely on predictions specific to a particular task. There is some circularity in adopting a different way of evaluating learning at the same time as proposing a different way of learning. Findings based on a simplistic outcome seem more secure, at least for this initial investigation.

Because user interface development is not a prescriptive science, there is a need to evaluate interface features reasonably quickly. The HCI literature refers to this part of software development as "usability testing". This experiment shows alternative UI characteristics (i.e. extensibility) being evaluated under similar constraints. Although the measures and treatments used here may not be ideal, experiment A demonstrates the potential of testing interfaces in terms other than usability. In other words, this kind of study is appropriate for developing the user interface and evaluating specific effects of user interface components.

To investigate the software in an actual classroom setting, in concert with the rest of the student's environment, a longitudinal study is desirable. Such a study on how to exploit the software would be a natural succession to this study on how to build software.

5.2. Experiment B:

Experiment B established that creating link semantics forced subjects to spend more effort dealing with the material in a global, comprehensive manner. Change in the type of verbalization is small compared to the shift towards scrutinizing global entities rather than details.

The verbalization dimension is an indication of the kind of thinking the subject is doing concerning their task. Although the treatment does produce more evaluation segments, subjects do not spend more time planning their designs. It appears that treatment subjects have to make more choices, but this does not mean they think more carefully about the task or consider strategy more often. The change in the task - requiring users to create link meanings - changed cognitive activity mainly by making subjects deal with the totality of the material.

This is gratifying to the extent that it shows that the treatment had the kind of effect that was intended. Encouraging students to reflect in global terms and on general themes clearly fits constructivist objectives. However, the theory of knowledge-as-design and the idea that the subject should "struggle for form" would be better satisfied by a corresponding increase in planning segments. If more planning activity could be produced, outcomes such as those found in experiment A might be even more pronounced.

Increasing planning type activity may require changing the conditions of the study. Another possibility is that planning activity is already saturated - that is, that a certain proportion of other activity (attention, evaluation and intention) is essential in order to gather information necessary to make coherent planning. If this is the case, a more sensitive instrument is needed to further discriminate the planning category, and distinguish the more beneficial design activities.

5.2.1. Link Meanings

The benefits of producing global thinking may have implications for the link semantics adopted by subjects. Topical symbol labels are generated in experiment A, such as "food" and "location" (see appendix E for others), drawn from the broad themes of the text. Other types of semantics, such as "part of" (i.e. the logical relations advocated by Holley & Dansereau, 1984b) , do not deal with the text as a whole, but rather with two particular elements of the text. It seems unlikely that these type of meanings would direct subjects to consider the general character of the material. They would correspond to the "relation" category in the coding scheme, rather than the "global" or "network" categories that were affected in experiment B. The use of such link meanings might produce a positive effect, but it is unclear how it would relate to effects observed in experiment B.

Another question worth considering is whether the value of the constructive process is reflected in the quality of the end product. This often appears as a tacit assumption in work on graphic organizers and similar devices. The final product is used to evaluate the student's understanding, and the focus is on teaching students to produce "good" representations. In fact, the best learning process might produce a rather dreadful representation. In terms of link meanings, for example, it may turn out that topical links are important to using hypertext authoring as a learning experience, but that these same link types are rather poor for producing a useful hypertext document. Certainly the documents resulting from experiments A and B indicate little about the process of their construction.

5.3. The software

The software was not intended as the best possible constructive environment, but rather one in which experimental contrasts could be effectively isolated with respect to particular attributes of the user interface. The construction and development of even this restricted environment has clarified a number of ideas from the framework.

The emphasis on producing a simple-to-learn environment has proved worthwhile. The original motivation for simplicity was to reduce variation in subjects' skill within the experiment; in other words, to simplify the experimental design. As it turned out, subjects had no problem operating the program, and only a short training session was necessary. This provides some support for the contention that constructive learning activities can be enhanced or created though computing environments without extra overhead in learning, such as required by programming in LOGO.

More specifically, the program serves as a counterexample to claims that hypertext concepts of "link" and "node" are difficult for students to come to terms with. (cf. Jonassen & Mandl, 1990). With obvious mechanisms and a constructive task, these concepts presented no difficulties for the adult subjects in the experiments. This success may have as much to do with interpretation of the linking mechanism as it has to do with the simplicity of the interface. Rather than expect users to conform to a particular interpretation of links and nodes, they are introduced to the linking mechanism and left to form their own interpretation.

5.4. Constructivist implications

The evidence indicates that the treatment did enrich constructive activity on the part of the subject. The additional effort of choosing an external representation (the link meanings) produced a difference in internal process (thinking in global terms). This is accompanied by a specific outcome: quantitative improvement in retention.

In a specific sense, the results can be attributed to design opportunities created by the treatment. The condition manipulated in the experiment compels the subjects to participate in a design act; they must create meanings for their links. Therefore these subjects are engaged in constructing a part of the representation that their counterparts in the control group do not construct.

The treatment also offers an opportunity to create any kind of link semantics the subject wants. However, subjects did not exploit a wide range of semantics in their choice of link meanings; rather they chose simple topical labels from the text content. This dimension of the "design opportunity" is not conveyed to the subjects. Their link semantics are no more creative than those of the control subjects. Training or additional support of some type may be able to expose a broader range of design possibilities to the users, and extract further benefit from the constructive character of the interface.

Benefits have been demonstrated only in the context of one particular interface. The framework of figure 2.4 establishes extensibility as the distinguishing property for software to be used in the creative manner of a medium. The notion of undetermined features provides the means to producing and identifying this property in specific tasks and techniques of other computer software.

5.5. Applying pedagogical roles

The ideas synopsized in figure 2.4 have implications beyond the few elements examined by experiments A and B. The broad intent of this framework is to prescribe software features (specifically user interface features) according to the way the software is to be used in the classroom environment. The most original, and therefore provocative idea was examined first: that undetermined features would support the role of software as a constructive medium.

User interfaces for the other roles of the framework have not been tested herein. Each role describes a different pedagogical context, so controlled experimental evaluation of these distinctions would be difficult. The experiments do show that the framework can be used in developing and evaluating an interface, making it feasible to construct an interface appropriate to a given pedagogical context. In future, such interfaces could be evaluated by software developers, teachers, and students.

Matching user interface considerations to the learning environment demands further refinement and experience in applying the framework. For example, students using challenging interfaces may require individual help from the teacher. Each student may be designing something unique, requiring extra time and effort on the part of the teacher to understand each separate creation. At this point it is unclear how the software might itself attempt to mitigate such problems. Indeed, such practical considerations might curtail the use of software to roles which are effective for group instruction or amenable to the particular instructor's teaching style.

Case studies offer one possibility for refining the framework - to study the use of software in different classroom situations and determine whether the framework correctly distinguishes important aspects of the interface. In the meantime, conclusions can be drawn concerning the specific features studied in this thesis.

5.5.1. Undetermined features

The idea of different user interface priorities for different pedagogical roles has been tempered by experience with this program. The idea that extensibility can be satisfied exclusively through the existence of particular features is not tenable. Since extensibility is a property of how software is used, the appropriate features must be accessible, understandable and relevant or they will not be used.

The experiments examined one extensible feature - underdetermined link symbols. It would be foolhardy to generalize to all sorts of other features on the basis of these results alone, since even the notion of the "extensible interface" has to be defined in operational terms - in terms of what screen elements users will find amenable to reinterpretation. The experiments do demonstrate that the user interface suggestions based on the framework do bear fruit, and produce positive effects as predicted.

In isolation, the experimental results may say more about the specific task of creating semantic labels than the more general notions about user interface design. Even though subjects explore very few alternatives in selecting meanings, making these constructions is shown to be more beneficial under the treatment, and the case for having one particular underdetermined feature is supported.

Other underdetermined features have yet to be examined. In the hypertext environment, an obvious candidate is the nodes. In the experiments, the nodes were treated simply as text pages, thus limiting the proliferation of variables. However, the grand vision of hypertext as a constructive environment, as portrayed in section 2.9.3, uses nodes as general placeholders for images, charts, footnotes, and so on. This leaves the user in charge of their semantics, much as do the boxes of BOXER. The effect this would have in conjunction with and separately from the undetermined anchors would be of interest.

Similar effects should also be reproducible in other environments, using other features. Sounds and colours could be used as undetermined features in conventional applications such as the word processor. When a number of such features and contexts have been examined, more definite statements will be available concerning the general utility of the framework and the concept of the "extensible interface". For now, it is satisfactory that undetermined features have proven viable as a component of a user interface.

5.6. Future Directions

A number of possible directions for this work have been mentioned already. This section summarizes the essential thrust of each suggestion.

The specific experiments could be extended by adding additional features or tasks to the conditions, or by allowing subjects to take their tasks to completion. Further support could be provided for constructive activity, such as encouraging subjects to think carefully about alternative link semantics. Outcomes more informative than free recall should be measured. This would provide a more thorough examination of the extensible interface, and at the same time help refine a hypertext system for practical use as a constructive environment.

Developing new interfaces and new ways of evaluating interface components is important to the broad claims included in this thesis. Case studies of software in current use would be ideal for fleshing out the framework, and new software should be built and evaluated using the principles of the framework. This would pursue the question of whether user interfaces can be specifically suited to the various roles.

Practical use of such interfaces would entail matching these roles to the teaching styles and instructional methods used in the classroom. Such a matching goes beyond the theoretical concerns of this work, involving such things as students' and teachers' beliefs and attitudes concerning the computing technology. Longitudinal studies would be necessary to establish how new systems could be integrated into an existing teaching and learning milieu. For example, experiments A and B isolated one very specific element in order to establish a positive effect. If many different aspects of the environment supported constructive activity, it might not be possible to say if this effect will be a relatively large or small contribution. This is a separate challenge, in a larger educational context - to define learning situations and tasks which exploit characteristics of different interfaces.

Scholastic standing and other substantial improvements could be confirmed by a long term study. Furthermore, the idea that processes may be different in the long term has already been raised - in the case of link semantics, where subjects could build on their skills and explore more design alternatives. Finally, concerns for ecological validity must eventually turn to placing the software in the hands of genuine practitioners.

5.7. Summary

The experiments have confirmed some predictions, but also have produced new insight into "undetermined features" and other constructs of the framework. Effects were not due to a wide range of designs produced by the students, so providing "design opportunities" now seems more subtle than simply providing the widest possible range of semantic interpretation. Training and long exposure may be needed to produce the most benefit from software as a "medium"; but with an appropriate interface, there are spontaneous effects in the users designing representations, even without a wide range of designs being considered. This is a gratifying achievement, since others have to measure progress in terms of semesters. Results are produced here in 15 minute sessions, with a simple to use program, and a straight forward task requiring almost no training.

Longitudinal studies are suggested for exploring how to incorporate these programs into the classroom. The main focus here has been what to put into these interfaces, which still needs to be evaluated on a feature-by-feature and program-by-program basis. For this purpose, controlled experiments and case studies offer a more effective use of time. The framework suggests how to interpret various features, and positive indications from a small study can be pursued by incorporating the same features into a longer study.

Ultimately, the objective is to build better interfaces. The fundamental idea is to make the user interface appropriate to how the software is used; that we can assist certain pedagogical approaches by providing appropriate interfaces. These experiments provide some initial clues on how to explore this potential. Exact prescriptions for the development of interfaces and their integration into pedagogical practice are some ways off, but these results are a step in that direction.

References

Anderson, J. R.,Boyle, C. F., & Yost, G. (1985). *The geometry tutor.* Proceedings, International Joint Conference on Artificial Intelligence, (pp. 1-7). Los Angeles: Morgan Kauffman.

Arnheim, R. (1969). Visual Thinking. Berkeley, CA: University of California Press.

Ausubel, D. P. (1963). The Psychology of Meaningful Verbal Learning. New York: Grune & Stratton.

Baecker, R. M., & Buxton, W. A. S. (Ed.). (1987). Human-Computer Interaction: A Multidisciplinary Approach. San Mateo, CA: Morgan Kaufmann.

Bannon, L. J., & Bødker, S. (1991). *Beyond the interface: encountering artifacts in use.* In J. M. Carroll (Eds.), Designing Interaction: Psychology at the Human-Computer Interface (pp. 227-253). Cambridge: Cambridge University Press.

Brooks, F. P., Jr. (1988). *Grasping reality through illusion - interactive graphics serving science.* Proceedings, Computer-Human Interaction, (pp. 1-11). ACM press.

Brown, A. L., & Palincsar, A. S. (1984). *Reciprocal teaching of comprehension-fostering and comprehension monitoring activities.* Cognition and Instruction, 1, 117-175.

Brown, J. S., Collins, A., & Duguid, P. (1989). *Situated cognition and the culture of learning.* Educational Researcher, 18(1), 32-42.

Bruner, J. S. (1966). *Some elements of discovery.* In L. S. Shulman & E. R. Keislar (Eds.), Learning by discovery: A critical approach. Chicago: Rand-McNally.

Cardinale, L. A. (1991). *Conceptual models and user interaction with computers.* Computers in Human Behaviour, 7, 163-169.

Carroll, J. M., & Kellogg, W. A. (1989). *Artifact as theory-nexus: hermaneutics meets theory-based design.* Proceedings, Computer-Human Interaction, (pp. 7-14). ACM Press.

Carroll, J. M., Mack, R. L., & Kellogg, W. A. (1988). *Interface metaphors and user interface design.* In M. Helender (Eds.), Handbook of Human-Computer Interaction (pp. 67-85). Elsevier Science Publishers (North Holland).

Clarke, J. H. (1990). Patterns of Thinking; Integrating Learning Skills in Content Thinking. Boston, Mass.: Allyn and Bacon.

Cohen, R., & Geva, E. (1989). *Designing LOGO-like environments for young children: the interaction between theory and practice.* Journal of Educational Computing Research, 5(3), 349-377.

Dickson, W. P. (1985). *Thought provoking software: juxtaposing symbol systems.* Educational Researcher(May 1985), 30-38.

diSessa, A. A. (1982). *Knowledge in pieces.* In G. Forman & P. B. Pufall (Eds.), Constructivism in the Computer Age (pp. 49-70). Hillsdale, NJ: Erlbaum.

diSessa, A. A. (1985). *A principled design for an integrated computational environment.* Human-Computer Interaction, 1(1), 1-47.

Ericsson, K. A., & Simon, A. S. (1984). Protocol Analysis: Verbal Reports as Data. Cambridge, Mass.: Bradford Books.

Falbel, A. (1990). *The computer as a convivial tool.* In I. Harel (Eds.), Constructionist Learning MIT: MIT media laboratory.

Fisher, K. M. e. a. (1988). *Computer-based knowledge representation as a tool for students and teachers.* Proceedings, Proceedings of the annual meeting of the AERA, New Orleans.

Fosnot, C. (1984). *Media and technology in education: a constructivist view.* ECJT, 32(4).

Gaver, W. (1991). *Technology affordances.* Proceedings, Computer-Human Interaction, (pp. 79-84). New Orleans: ACM press.

Goetz, E. T. (1984). *The role of spatial strategies in processing and remembering text.* In C. D. Holley & D. F. Dansereau (Eds.), Spatial Learning Strategies (pp. 47-77). Orlando: Academic Press.

Gordon, S.,Gustavel, J.,Moore, J., & Hankey, J. (1988). *The effect of hypertext on reader knowledge representation.* Proceedings, Human Factors Society 32nd Annual Meeting, (pp. 296-300).

Gray, S. H. (1990). *Using protocol analyses and drawings to study mental models during hypertext navigation.* International Journal of Human-Computer Interaction, 2(4).

Greif, S. (1991). *The role of german work psychology in the design of artifacts.* In J. M. Carroll (Eds.), Designing Interaction: Psychology at the Human-Computer Interface Cambridge: Cambridge University Press.

Harel, I., & Papert, S. (1990). *Software design as a learning environment.* In I. Harel (Eds.), Constructionist Learning MIT: MIT media laboratory.

Hollan, J. D.,Hutchins, E. L., & Weitzman, L. (1984). *STEAMER: an interactive inspectable simulation-based training system.* AI Magazine, 5(2), 15-27.

Holley, C. D., & Dansereau, D. F. (1984a). *The development of spatial learning strategies.* In C. D. Holley & D. F. Dansereau (Eds.), Spatial Learning Strategies (pp. 3-20). Orlando: Academic Press.

Holley, C. D., & Dansereau, D. F. (1984b). *Networking: Technique and the empirical evidence.* In C. D. Holley & D. F. Dansereau (Eds.), Spatial Learning Strategies (pp. 81-108). Orlando: Academic Press.

Holley, C. D., & Dansereau, D. F. (Ed.). (1984c). Spatial Learning Strategies. Orlando: Academic Press.

Jonassen, D. H., & Mandl, H. (Ed.). (1990). Designing Hypermedia for Learning. New York: Springer-Verlag.

Kay, A. (1987). *Doing with images makes symbols: communicating with computers.* In Industry Leaders in Computer Science, Distinguished Lecture Series. Stanford: University Video Communications.

Kintsch, W. (1977). Memory and Cognition. New York: Wiley.

Kozma, R. B. (1991). *Learning with Media.* Review of Educational Research, 61(2), 179-211.

Loyd, B. H., & Steele, J. L. (1986). *Assessment of Reading Comprehension: A Comparison of Constructs.* Reading Psychology, 7(1), 1-10.

MacLean, A.,Bellotti, V.,Young, R., & Moran, T. (1991). *Reaching through analogy: a design rationale perspective on roles of analogy.* Proceedings, Computer-Human Interaction, (pp. 79-84). New Orleans: ACM press.
MacLean, A.,Carter, K.,Lövstrand, L., & Moran, T. (1990). *User-tailorable systems: pressing the issue with buttons.* Proceedings, Computer-Human Interaction, (pp. 175-182). Seattle, Washington: ACM press.
McKeachie, W. J. (1984). *Spatial strategies: critique and educational implications.* In C. D. Holley & D. F. Dansereau (Eds.), Spatial Learning Strategies (pp. 301-312). Orlando: Academic Press.
Meyer, B. (1975). The Organization of Prose and its Effects on Memory. New York: American Elsevier.
Nielsen, J. (1991). Hypertext and Hypermedia. San Diego: Academic Press.
Norman, D. (1991). *Cognitive artifacts.* In J. M. Carroll (Eds.), Designing Interaction: Psychology at the Human-Computer Interface (pp. 17-38). Cambridge: Cambridge University Press.
Norman, D. A. (1983). *Some observations on mental models.* In A. L. Stevens & D. Getner (Eds.), Mental Models Hillsdale, N.J.: Lawrence Erlbaum.
Norman, D. A. (1988). The Psychology of Everyday Things. New York: Basic Books.
Novak, J. D., & Gowin, D. B. (1984). Learning How To Learn. Cambridge: Cambridge University Press.
Papert, S. (1980). Mindstorms: Children, Computers and Powerful Ideas. New York: Basic Books.
Papert, S. (1990). Introduction. In I. Harel (Eds.), Constructionist Learning MIT: MIT media laboratory.
Perkins, D. (1986). Knowledge as Design. Hillsdale, New Jersey: Lawrence Erlbaum.
Piaget, J. (1950). The Psychology of Intelligence. New York: Harcourt, Brace.
Quillian, M. R. (1968). *Semantic meaning.* In M. Minsky (Eds.), Semantic Information Processing Cambridge, MA: MIT Press.
Quillian, M. R. (1969). *The teachable language comprehender.* Communications of the ACM, 12, 459-475.
Rao, U., & Turoff, M. (1990). *Hypertext functionality: a theoretical framework.* International Journal of Human-Computer Interaction, 2(4), 333-357.
Reynolds, S., & Dansereau, D. F. (1990). *The knowledge hypermap: an alternative to hypertext.* Computers in Education, 14(5), 409-416.
Rueff, S. J. (1992). *Pictures, poseurs, and posers: the power of scientific visualization and the questions it raises.* Computer Graphics, 26(3), 202-204.
Scardamalia, M., & Bereiter, C. (1991). *Higher Levels of Agency for Children in Knowledge Building: A Challenge for the Design of New Knowledge Media.* Journal of the Learning Sciences, 1(1), 37-68.
Shneiderman, B. (1983). *Direct manipulation: a step beyond programming languages.* IEEE Computer, 16(8), 67-69.
Shneiderman, B. (1987). Designing the User Interface. Reading, Mass: Addison-Wesley.

Soloman, G. (1988). *AI in reverse: computer tools that turn cognitive.* Journal of Educational Computing Research, 4(2), 123-134.
Teshiba, K., & Chignell, M. (1988). Development of a user-model evaluation technique for hypermedia based interfaces (Working paper No. 88-15). USC.
Thompson, P. (1987). *Mathematical microworlds and ICAI.* Proceedings, Workshop on ICAI and Personal Computers, San Diego.
Thorndyke (1977). *Cognitive structures in the comprehension of narrative text.* Conitive Psychology, 9, 77-110.
Tiessen, E. L. (1991). The Icon Assignment (Unpublished Manuscript) University of Toronto.
Tukey, J.W. (1977). Exploratory Data Analysis. Reading, Mass.: Addison-Wesley.
White, B. Y. (1988). Thinker-tools: causal models, conceptual change, and science education Washington, D.C: The National Science Foundation.

Appendix A: Consent form for experimental subjects

Consent Form

Title: A pedagogical role for conceptual models in user interface design
Ph.D. Research
Candidate : Edward Brown Supervisor: Dr. R. S. McLean

Thank you for helping with this work. In this study you will perform a task with some text using a computer program. The purpose is to find how performing this task will affect people's memory of the text.

You will be shown how to use a computer program. Following this, you will be asked to read and manipulate the text using the computer program. Finally you will attempt to recall as much of the text as you can.

All of your activities, comments, and responses will be kept strictly confidential. Your name will not appear on any reports, or be mentioned in connection with this work. You will not be graded on your performance, nor will information from this study be made available for any purposes of evaluation.

If, for some reason, you decide to withdraw from this study, you are free to do so at any time.

I have read and understand the information on this consent form and I agree to participate in this research.

Signature: ______________________________
Date:___________

Appendix B: Subject instructions for experiment A

Step 1: Introduction and training task	
Spoken Instructions	**Action**
This session will take approximately 90 minutes. You must be available for 30 minutes at the same time next week, after which you will be paid $20 for participation in this study. Please fill out this consent form and supply the information on the reverse side: name, age, sex, level/type of education.	*Preparing subjects* • Fill out consent form
In this study I am interested in whether connecting up ideas in text will help people remember those ideas. We will have you connecting up ideas and then find out how much you can remember. We have a special computer program for connecting up ideas, which you will use. This session is divided into three parts. The first part is to learn how this program works. {introduce mouse, cursor pointing & clicking} This is what the text looks like. There are pages on the screen just like pages on a desk or table. These pages have text on them. In this program, ideas are linked together by small symbols which we call "anchors". If you press on an anchor, it takes you to another idea on another page. Ideas on these two pages are "linked" by this pair of anchors. If we press on the other anchor, it takes us back to the starting place. So these two places are connected by a link. There are now two pages on the screen. These pages can be moved about by dragging on the top bar of the page, where the page number appears. {how to drag with the mouse}	*Introducing the study* *Demo of simple features* • Open program to first page • Press anchor for 2nd page • Press on anchor; return to 1st page • Move page
Try the program at your own station. Press on an anchor, and try moving pages around.	*Practice simple features* • Subject uses mouse • Subject tries anchor;drags page
You see there are different kinds of anchors. This is because there are different reasons for linking ideas together. Each reason or type of link uses a different symbol. You now have a training task to get used to the program. This is not part of the experiment task, it simply gets you used to using the program. In this task, I want you to use the links to find ideas in the text. The ideas you must find are given by a question at the bottom of the screen.	•Point out icon palette *Introducing the training task* •Point out questions.

When you find an answer to a question, you press on the button under the question. For example, to answer this question {read the question: "what part of the bird is the supercillium?"} we must look for the correct idea. Reading through this page, I encounter the word "supercillium", but no explanation of the word. I choose this anchor because {explain how it relates to the question: it is on the word "supercillium", which is what the question is about.} This takes us to a new page, where we find the idea relating to this question. So now I can say I have found it. Now a new question has appeared. Here's what happens if I don't find the correct idea in the text. If you can't find the idea, you can skip to the next question with this button. However, you might have to try several times to find the right idea. You now have to try this on your own. You will have five minutes to find as many ideas as you can. Remember, this is just to get used to the program, so don't worry about how well you perform.	•read question •follow link •press found it button •return to incorrect page; press button •point out skip button
	Perform training task
You should now understand that anchors and links can be used to move from page to page, and to connect ideas in the text. You may NOT have noticed that each anchor symbol was used for a different meaning. You may not have discovered what these meanings were, but you should understand that they could be helpful in trying to find you way through the text. Notice that each symbol was used for many different links.	*Review training task*

Step 2: Linking Task (Treatment condition)	
Spoken Instructions	**Action**
Part {2,3} of this session requires you to make links yourself. I will now show you how to do this. Choose the anchor you want from here. Notice you now have a stamp pad. Use this to stamp the anchor onto the page where you want the anchor to appear. The anchor will beep until you stamp at the other idea to be connected. Then the link is complete. You can also adjust the anchor position one it is placed, by dragging it. Since you will start with no links, you have to move through the pages one by one, using the arrow keys.	*How to make links* •Practice Screen •Make a link •Drag anchor •Arrow Keys
I will now make some additional links so you can see a few more examples.	*Model link making* •create more links
Practice these features at your station. Use the arrow keys to move through pages, create some links and drag them around. Create at least 5 different links and drag them.	*Practice Link skills* •Subject arrow keys •Subject makes link •Subject drags link

[Control condition only] You are now ready to do the experiment task. In this task, you will be making your own anchors. Your task is to connect ideas on different pages together so people could find ideas by using the anchors. This could be similar to how you used the anchors when you were answering questions in the training task. Now you will be making the links instead of answering questions. I will be giving you linking ideas to use. Use one anchor symbol for each linking meaning. {In other words, the same anchor symbol is used when the reason for linking facts together is similar, and different symbols when the reason for linking facts together is different.} There are 12 symbols available, but most people only need to use a few of them. You should find yourself using some symbols for many links, when it is a recurring linking idea. On the other hand, you might use other symbol-links only once. You can decide what meanings to choose as you work. Many people find it convenient to refer to a list of meanings for each symbol. You will be given this list of meanings before you start. Don't simply link up individual words, but look for complete facts or ideas in the text. You have 15 minutes to make as many links as you like. You will have 10 new pages to work with. These pages will appear one by one automatically for you to read. Once all pages have appeared, you can examine the link meanings and begin linking. Any questions?	*Introduction to link task* •Give symbol sheet to subject.
[Treatment condition only] You are now ready to do the experiment task. In this task, you will be making your own anchors. Your task is to connect ideas on different pages together so people could find ideas by using the anchors. This could be similar to how you used the anchors when you were answering questions in the training task. Now you will be making the links instead of answering questions. You must decide what the main linking ideas are in the text. Use one anchor symbol for each linking meaning. {In other words, use the same anchor symbol when the reason for linking facts together is similar, and different symbols when the reason for linking facts together is different.} There are 12 symbols available, but most people only need to use a few of them. You should find yourself using some symbols for many links, when you have a recurring linking idea. On the other hand, you might use other symbol-links only once. Of course, you can add new linking meanings as you work. Many people find it write down the meanings they use for each symbol. Here is a sheet on which you can do this, if you like. Don't simply link up individual words, but look for complete facts or ideas in the text. You have 15 minutes to make as many links as you like. You will have 10 new pages to work with. These pages will appear one by one automatically for you to read. Once all pages have appeared, you can begin linking. Any questions?	*Introduction to link task* •Give symbol sheet to subject.

You may begin now Time has expired.	*Performance of link task* •connect earphones; •open create Screen • start 15 minute timer

Step 3: Immediate Recall	
Spoken Instructions	**Action**
Now we will see how much you can remember. Recall as many ideas from the pages IN AS MUCH DETAIL as you can. Report all the ideas you can from the text, in whatever order they occur to you. You may put a line through mistakes, but don't erase anything you write. Details are important, so be sure to report any details you can recall.	*Recall Longhand* •provide pen, paper

Step 4: Instructions for reading task	
Spoken Instructions	**Action**
In part {2,3} of this study we want to know how you remember when you simply read some text. This will provide a basis of comparison with the experiment task of actually making some links. You will study the text, and then you will be asked to recall it. Since you will have no links, you have to move through the pages one by one, using the arrow keys. {Try it, if you haven't already}. You have 15 minutes to read as you like. You will have 10 new pages to work with. These pages will appear one by one automatically for you to read. Once all pages have appeared, you can look through the pages as you like. Any questions?	*Introducing the reading task* •Return to practice screen
You may begin now Time has expired.	*Performance of read task* •open read Screen • start 15 minute timer

Step 5: Immediate Recall	
Spoken Instructions	**Action**
Now we will see how much you can remember. Recall as many ideas from the pages IN AS MUCH DETAIL as you can. Report all the ideas you can from the text, in whatever order they occur to you. You may put a line through mistakes, but don't erase anything you write. Details are important, so be sure to report any details you can recall.	*Recall Longhand* •provide pen, paper

Step 6: Delayed Recall	
Spoken Instructions	**Action**
The topic you looked at first was {mountain gorillas, polar bears}. Now we will see how much you can remember. Recall as many ideas from the pages IN AS MUCH DETAIL as you can. Report all the ideas you can from the text, in whatever order they occur to you. You may put a line through mistakes, but don't erase anything you write. Details are important, so be sure to report any details you can recall.	*Recall Longhand* •provide pen, paper

Step 7: Delayed Recall	
Spoken Instructions	**Action**
The topic you looked at second was {mountain gorillas, polar bears}. Now we will see how much you can remember. Recall as many ideas from the pages IN AS MUCH DETAIL as you can. Report all the ideas you can from the text, in whatever order they occur to you. You may put a line through mistakes, but don't erase anything you write. Details are important, so be sure to report any details you can recall.	*Recall Longhand* •provide pen, paper

Appendix C: Text of experimental hypertext materials

The text content of the two hypertext documents used in the experimental tasks are given below. A slash "/" separates text segments used to identify different "idea units" in data analysis. When experimental subjects view the nodes, they do not contain these slashes.

	Page Set 1
Page 1	Recent counts show / that there are fewer than 400 mountain gorillas remaining./ Much of the forest upon which they depend for food and shelter/ has been cut down./ With so few individuals left,/ the species can be considered a prime candidate for extinction./
Page 2	Gorillas in the wild are found only in Africa./ Mountain gorillas live in a small area of bamboo and mountain forests/ on the slopes of the Virunga Volcanos/ in Rwanda, Zaire and Uganda./ They are the rarest of the three subspecies of gorilla./
Page 3	Mountain gorillas are slaughtered deliberately/ to make souvenirs./ Adult gorillas are found with their heads and hands missing./ Sometimes they are killed accidentally/ in the traps/ that poachers set for antelope./
Page 4	Gorilla hands and feet have been sold/ for about 25 dollars each/ as novelty ashtrays./ Locally, witch doctors use ears, tongues, genitals and small fingers/ to make a potion/ which is supposed to give the strength of a gorilla./ This trade is illegal./
Page 5	As more and more people need space in which to live,/ they are chopping down the mountain forests./ Land is also cleared/ to graze cattle/ and grow crops./ In 1969,/ nearly half of the park where the mountains gorillas live was given to farmers./
Page 6	In Rwanda, anti-poaching patrols/ have reduced the killing of adult gorillas and smuggling of babies./ Each ranger finding a wire or hemp snare is paid a bonus./ Poachers can still enter the park from Zaire or Uganda/ and cross the boarder into Rwanda to set traps./
Page 7	In 1978 the Mountain Gorilla Project was set up/ in an effort to save the gorilla from extinction./ This project works with the Rwandan government/ to provide equipment, training, and advice to anti-poaching patrols./ It also teaches local people about the gorillas./
Page 8	There are no mountain gorillas in zoos, only lowland ones./ Most of these are now bred in captivity./ In the better zoos, the gorillas live in compounds/ where they can move about and forage as they would naturally./
Page 9	Gorillas take time to forage and digest their food./ A family group's home range may be 5-30 square kilometers./ A gorilla group does not strip all the vegetation away from an eating site before moving on;/ they leave plenty of shoots that can grow up again./
Page 10	Most births occur at night/ in the mother's night nest./ A baby will crawl on its belly at nine weeks,/ and walks on two legs by twelve months./ About fourty percent of babies die before age three.

The text content of the second hypertext document used in the experimental tasks is given below. A slash "/" separates text segments used to identify different "idea units" in data analysis. When experimental subjects view the nodes, they do not contain these slashes.

	Page Set 2
Page 1	All polar bears migrate./ For example, in winter the Greenland population heads south/ across the frozen sea between Greenland and Iceland/ in search of food./ The bears stay within 200 kilometers of their population center./
Page 2	The Inuit traditionally hunted polar bears for survival./ They made bones into kitchen utensils,/ hides into clothing,/ and teeth and claws into jewellery./ Today hunters must obtain licences./
Page 3	In order to identify the furs of lawfully killed polar bears,/ Canadian Inuit are issued a quota of tags/ at the start of the winter./ When a hunter kills a bear,/ he attaches a tag to its pelt./ Only pelts with tags can be sold./
Page 4	Churchill, Canada is called the polar bear capital of the world./ Churchill is on the bears' migration route./ In autumn, the bears leave the forests,/ and return to the sea/ to hunt for seals./
Page 5	In the town of Churchill, the bears often come to visit the town's garbage dump/ and sometimes come into the town itself./ They have even killed several people/ in the town./ Bears that damage property are tranquillized/ and moved to a compound/ until the bay has frozen./
Page 6	Researchers want to track the movements of polar bears./ They attach tags to the bear's ears/ or punch a permanent code onto its lip./ A few bears are fitted with radio transmitters/ which can be picked up by satellites or directional receivers./
Page 7	The sea-ice of the Arctic Ocean is in constant motion./ Consequently, a polar bear cannot retain and defend a distinct piece of territory./ Bears without cubs travel alone,/ only congregating where food is particularly abundant./
Page 8	The polar bear eats mostly meat./ It drinks little water,/ and is very lazy after eating large meals/ at irregular times./ The weather affects the distribution of seals,/ the polar bear's primary food./ When conditions are favorable,/ the bear searches constantly for its next meal./
Page 9	In late spring seal bask on the ice,/ and bears stalk them slowly./ If a seal wakes the bear stops/ and blends itself into the background./ Bears have even been known to cover up the black tip on their nose/ with a paw./
Page 10	Unlike other adults, a pregnant female does not roam the Arctic during the winter./ Instead, she digs a large den in a snowdrift./ Cubs are born in December,/ and grow to the size of a man in one year.

The text content of the hypertext document used for the training task (Step 1: habituation to the software environment described in section 3.4.3) is given below. No slashes are present, since this text was not divided into text segments.

	Training set
Page 1	The study and naming of the bird's external features (bill, eyes, wings, feet, and so on) is called topography. Knowledge of bird topography is essential for producing accurate physical descriptions of birds, to categorize new species or to properly identify an unknown bird in the field. Field identification depends heavily on descriptions such as "males have a buff-colored supercillium and dark red streaking in the malar region".
Page 2	In birds of prey the iris is often quite colorful. The color of a bird's iris can sometimes tell you roughly how old the bird is. In this species, the Sharp-shinned Hawk, young birds have bright yellow irises, while fully adult Sharp-shins have deep red or orange irises.
Page 3	In many groups such as hawks and parrots the nasal area is covered with a hard waxy surface called the CERE. The color of the cere often indicates the age of the bird. If this area of the bill is soft and fleshy (as in pigeons) the structure is called an OPERCULUM. The operculum is often folded over the nostril opening, partially or completely hiding it.
Page 4	Like other members of its family (the Fringillidae) the Grasshopper sparrow has a heavy conical seed-cracking bill. In sparrows the bill is well adapted to an omnivorous diet of seeds, insects and small fruits.
Page 5	In birds like sparrows that have distinctive stripes on the crown of their heads the individual stripes may be referred to as the "eyebrow" area (the supercillium), the eyeline (running behind the eye), and the median stripe (a central dark or light stripe running over the middle of the crown of the head). Mainly these terms are a convenience for describing the coloration of a particular species.
Page 6	The bill is one of the most obvious and interesting physical characteristics of birds, and can tell you a great deal about how the bird feeds and makes its living in the world. The Great Blue Heron shown here has a long stabbing bill adapted for capturing small fish, amphibians and invertebrates in marshes and other aquatic environments. It occasionally takes small land animals and rodents as well.
Page 7	Both the White Ibis (Order Ciconiiformes, Family Threskiornithidae) and the American Avocet (Order Charadriiformes, Family Recurvirostridae) have evolved long, delicate bills suited for picking up small invertebrates and other animal life from the shallow marine and freshwater areas where both species live. . Both ibises and avocets also have long legs adapted for wading in shallow water.
Page 8	Camouflage plummages offer an important means of protection to the chicks of birds like the Roseate Tern, which nests in relatively open ground nests on beaches and islands. The young birds have a drab brownish down that blends almost perfectly with the dirt, gravel and sand that usually surrounds their nests.

	Training set (continued)
Page 9	The environmental conditions under which an animal lives largely determine what physical and behavioral features will be favored in future generations of the species. Despite its ungainly looks the Brown Pelican is superbly adapted to fishing in marine coastal waters. With its long wings built for soaring over the water in search of Menhaden schools and the famous bill pouch for capturing fish under water the pelican is an efficient fish-catching machine.
Page 10	Note that many forest and thicket birds like the Blue Jay have short, thick wings and long tails. This combination gives the bird great maneuverability and allows fast flight acceleration from a standing start—both are essential qualities for a bird that lives among leaves and branches.
Page 11	The VANE is the main functional structure of the feather, providing (depending on location) either an aerodynamic surface or warmth and protection to the skin. In fact, most feathers do both jobs. The outer-most portion of the vane (the pennaceous portion) forms a strong, smooth surface suitable for flight and the inner vane closest to the skin (the plumulaceous portion) provides a soft, down-like insulation layer.
Page 12	Although we tend to think of feathers as flight-related, contour feathers are probably much older than flight feathers. Paleontologists now think feathers were developed by the ancestors of birds mainly to conserve body heat, not for any flight-related reason. It was only later that feathers became more specialized and adapted for flight.
Page 13	Down feathers are specially modified to trap air near the skin, underneath the protective contour feathers above. This layer of trapped air is warmed by the body and is actually what keeps the bird warm—the down feathers just keep the air in place and keep the airspace from collapsing. In adult birds down is most heavily concentrated on the breast and back.
Page 14	Males usually show the most distinctive secondary sexual characteristics, such as brightly colored plummage, colored skin patches, and enlarged or modified feather patterns. Females tend to be less brightly colored, for the protective camouflaging effect of drab plummage is crucial to prevent discovery of the nest site.
Page 15	Because the feathers covering the under surface of the wing have soft indistinct edges the whole area is collectively referred to as the WING LINING, rather than distinguishing each row of coverts. The primary remiges (flight feathers) of the wing are the main source of lift in flight.
Page 16	The lighter gray areas in this diagram represent the zones of lower pressure over the wing created by the airfoil. As the air flows over it curve of the wing's upper surface it must speed up. As it speeds up it must lose some pressure (Bernoulli's Law). The wing rises upward into this zone of lower pressure, creating lift.

Main sources for the information contained in the hypertext documents were:

Bright, Michael (1989). Project Wildlife, London: Fraklin Watts.

Kilpatrick, C. and Hard, J. (1986). The Hamlyn Animal Encyclopedia, London: Hamlyn Publishing.

Lynch, P. (1988) The Surface Anatomy of Birds, electronic document, Yale University School of Medicine.

Appendix D: Subject instructions for experiment B

Step 1A: Introduction and training task	
Spoken Instructions	**Action**
This experiment will take approximately 90 minutes. Please fill out this consent form and supply the information on the reverse side: name, age, sex, level/type of education.	*Preparing subjects* • Fill out consent form
In this study I am interested in how people connect up ideas in text. We will have you connecting up ideas and keep track of what you are doing. We have a special computer program for connecting up ideas, which you will use. This session is divided into three parts. The first part is to learn how this program works. This is what the text looks like. There are pages on the screen just like pages on a desk or table. These pages have text on them. In this program, ideas are linked together by small symbols which we call "anchors". If you press on an anchor, it takes you to another idea on another page. Ideas on these two pages are "linked" by this pair of anchors. If we press on the other anchor, it takes us back to the starting place. So these two places are connected by a link. There are now two pages on the screen. These pages can be moved about by dragging on the top bar of the page, where the page number appears.	*Introducing the study* •start video *Demo of simple features* • Open program to first page • Press anchor for 2nd page • Press on anchor; return to 1st page • Move page
Try the program at your own station. Press on an anchor, and try moving pages around.	*Practice simple features* • Subject uses mouse • Subject tries anchor;drags page
You see there are different kinds of anchors. This is because there are different reasons for linking ideas together. Each reason or type of link uses a different symbol. You now have a training task to get used to the program. This is not part of the experiment task, it simply gets you used to using the program. In this task, I want you to use the links to find ideas in the text. The ideas you must find are given by a question at the bottom of the screen.	•Point out icon palette *Introducing the training task* •Point out questions.

When you find an answer to a question, you press on the button under the question. For example, to answer this question {read the question: "what part of the bird is the supercillium?"} we must look for the correct idea. Reading through this page, I encounter the word "supercillium", but no explanation of the word. I choose this anchor because {explain how it relates to the question: it is on the word "supercillium", which is what the question is about.} This takes us to a new page, where we find the idea relating to this question. So now I can say I have found it. Now a new question has appeared. Here's what happens if I don't find the correct idea in the text. If you can't find the idea, you can skip to the next question with this button. However, you might have to try several times to find the right idea. You now have to try this on your own. You will have five minutes to find as many ideas as you can. Remember, this is just to get used to the program, so don't worry about how well you perform.	•read question •follow link •press found it button •return to incorrect page; press button •point out skip button
	Perform training task
You should now understand that anchors and links can be used to move from page to page, and to connect ideas in the text. You may NOT have noticed that each anchor symbol was used for a different meaning. You may not have discovered what these meanings were, but you should understand that they could be helpful in trying to find you way through the text. Notice that each symbol was used for many different links.	*Review training task*

Step 1B: Training for Verbal Protocols	
Spoken Instructions	**Action**
Now you understand how ideas are connected, it is time for you to make some of your own. Here is how you do this. Choose the anchor you want from here. Notice you now have a stamp pad. Use this to stamp the anchor onto the page where you want the anchor to appear. The anchor will beep until you stamp at the other idea to be connected. Then the link is complete. You can also adjust the anchor position one it is placed, by dragging it. Since you will start with no links, you have to move through the pages one by one, using the arrow keys.	*How to make links* •Practice Screen •Make a link •Drag anchor •Arrow Keys
I will now make some additional links so you can see a few more examples.	*Model link making* •create more links
Practice these features at your station. Use the arrow keys to move through pages, create some links and drag them around. Create at least 5 different links and drag them.	*Practice Link skills* •Subject arrow keys •Subject makes link •Subject drags link

Think and reason aloud, and report everything that passes through your mind while you are working. I am mainly interested in what you are thinking, so don't try to explain what you are doing, and don't plan what to say. Simply let your thoughts speak out loud. This means that you should be speaking most of the time. When you are reading, you should read out loud, too.	*Think-Aloud instructions*
We will try this out briefly before beginning the real experiment. We will use just a few pages, just to get the idea. Suggested prompts/corrections: • tell me what you're thinking • tell me more • now you should be saying what is in your mind • read out loud • report what you're thinking • you didn't say what were you thinking when you did that • use the same symbol for the same kind of link • I want to know what different links people use • what do you think as you are picking a symbol • what do you think as you are looking through the text	*Practice think-aloud* •open practice pages •subject links & speaks •correct subject

Step 2: Linking Task	
Spoken Instructions	**Action**
[Control condition only] You are now ready to do the experiment task. In this task, you will be making your own anchors. Your task is to connect ideas on different pages together so people could find ideas by using the anchors. This could be similar to how you used the anchors when you were answering questions in the training task. Now you will be making the links instead of answering questions. I will be giving you linking ideas to use. Use one anchor symbol for each linking meaning. {In other words, the same anchor symbol is used when the reason for linking facts together is similar, and different symbols when the reason for linking facts together is different.} There are 12 symbols available, but most people only need to use a few of them. You should find yourself using some symbols for many links, when it is a recurring linking idea. On the other hand, you might use other symbol-links only once. You can decide what meanings to choose as you work. Many people find it convenient to refer to a list of meanings for each symbol. You will be given this list of meanings before you start. Don't simply link up individual words, but look for complete facts or ideas in the text. Most importantly, remember to think out loud. You have 30 minutes to make as many links as you like. You will have 10 new pages to work with. Any questions?	*Introduction to link task* •Give symbol sheet to subject.

[Treatment condition only] You are now ready to do the experiment task. In this task, you will be making your own anchors. Your task is to connect ideas on different pages together so people could find ideas by using the anchors. This could be similar to how you used the anchors when you were answering questions in the training task. Now you will be making the links instead of answering questions. You must decide what the main linking ideas are in the text. Use one anchor symbol for each linking meaning. {In other words, use the same anchor symbol when the reason for linking facts together is similar, and different symbols when the reason for linking facts together is different.} There are 12 symbols available, but most people only need to use a few of them. You should find yourself using some symbols for many links, when you have a recurring linking idea. On the other hand, you might use other symbol-links only once. Of course, you can add new linking meanings as you work. Many people find it write down the meanings they use for each symbol. Here is a sheet on which you can do this, if you like. Don't simply link up individual words, but look for complete facts or ideas in the text. Most importantly, remember to think out loud. You have 30 minutes to make as many links as you like. You will have 10 new pages to work with. Any questions?	*Introduction to link task* •Give symbol sheet to subject.
You may begin now Suggested prompts: • tell me what are you thinking • say what is going through your mind • tell me more • read out loud Time has expired.	*Performance of link task* •open create Screen • start 30 minute timer

Appendix E: Frequency counts of link meanings.

Page set 1

Treatment group		Control group	
extinction	17	extinction	16
habitat & location	17	habitat & location	7
poaching	13	poaching	13
food	12	food	18
numbers & statistics	12	numbers & statistics	10
protection	7	protection	7
why killing	4	evidence of	3
dates	2	causes	2
how they die	1	part of	1
interaction with humans	1	kind of	1
subspecies of gorillas	1	is like	1
traps	1	has property	0
wild	1		
Total	89	Total	79

Page set 2

Treatment group		Control group	
food	20	food	12
migration	16	migration	8
territory	10	territory	12
seasons	8	seasons	5
licensing	8	licensing	6
hunting	7	hunting	10
similar meaning	5	causes	5
cubs	2	part of	2
Churchill	1	kind of	1
claws/lip	1	is like	1
cubs role in polar bear history	1	has property	1
frozen sea	1	evidence of	0
inuit/adult	1		
living habits	1		
population	1		
use of tags	1		
NO MEANING PROVIDED	1		
Total	85	Total	63

Notes: The numbers reflect occurrences of each meaning across all subjects for the linking task. Meanings on the left are those generated by treatment subjects, whereas those to the right were provided to control subjects as a list of choices. A meaning created by an individual treatment subject may vary slightly from the wording used in the above lists to collate these numbers. The shaded entries are those selected from the treatment results for use in the list for the control condition.

Appendix F: Recall scores from experiment A.

	Page Set Linked	Task Order	Link Recall Immed	Link Recall Delayed	Study Recall Immed	Study Recall Delayed
Control Subjects	Set 1	Study First	7	4	*16*	*8*
			9	1	*32*	*14*
			11	6	*16*	*7*
			21	16	*37*	*21*
		Link First	5	4	*31*	*27*
			15	5	*35*	*24*
			1	2	*8*	*3*
			10	7	*30*	*28*
	Set 2	Study First	*27*	*19*	39	20
			24	*13*	34	14
			18	*11*	37	13
			26	*13*	36	22
		Link First	*15*	*9*	11	14
			7	*6*	17	11
			36	*32*	52	41
			5	*4*	13	7
Treatment Subjects	Set 1	Study First	15	9	*37*	*7*
			9	6	*33*	*9*
			17	18	*30*	*19*
			13	7	*23*	*8*
		Link First	28	26	*39*	*37*
			12	7	*32*	*9*
			16	6	*30*	*1*
			20	18	*32*	*21*
	Set 2	Study First	*11*	*8*	21	10
			2	*0*	19	6
			13	*4*	22	3
			34	*26*	19	16
		Link First	*32*	*23*	46	34
			8	*4*	18	6
			14	*12*	29	18
			20	*17*	30	16

Note: Each line of the table shows the four recall scores for an individual subject. Recall for page set 2 is shown in italics to distinguish it from page set 1.

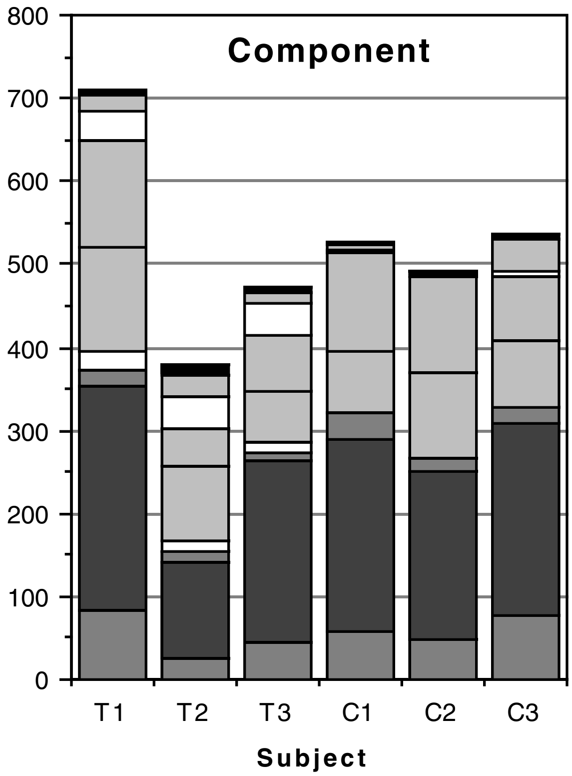
Appendix G: Sample segments to illustrate coding categories.

Verbalization Categories	
Category	**Sample Segment**
Intention	So I'll use a square for that...
Attention	Now there are certain screens that have information as to why forests are cut down.
Evaluation	which is kind of restrictive..
Planning	So maybe I should connect up page two and page ten somehow..

Component Categories	
Category	**Sample Segment**
Content	It says here that the species can be considered a prime candidate for extinction and..
Link	so that would be linked to that one ..
Type	I don't know if any of these symbols really mean that much
Network	and you can only get at them from the top page,
Operation	The beeping is annoying.
Meta-Comment	That's fresh in my mind still.
Relationship	I'm now looking to see if there's any other reason as to why forests are being cut down.
Global	First of all we have to find a common theme here.

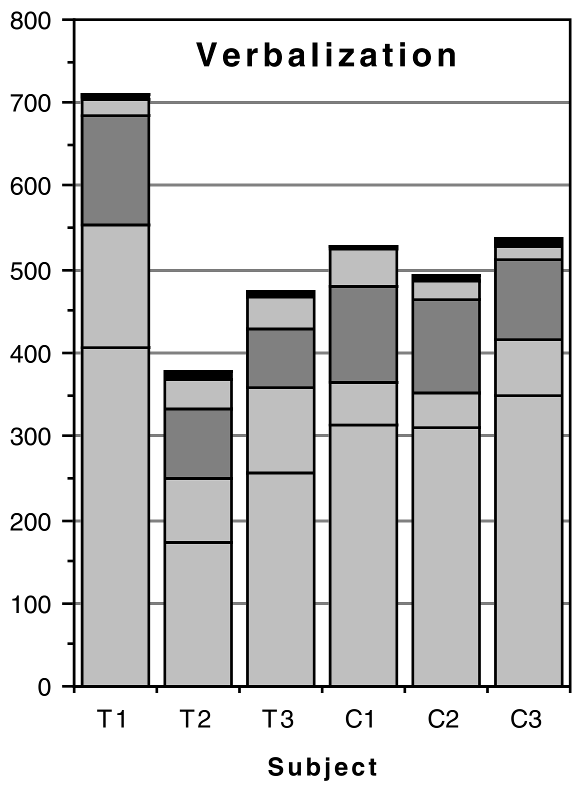
Note: all segments are coded on both dimensions. For example the sample segment for *relationship* on the component dimension is also coded *intention* on the verbalization dimension.

Appendix H: Frequency counts from experiment B.



Component Segments

Category	Treatment T1	Treatment T2	Treatment T3	Control C1	Control C2	Control C3
Unknown	7	12	7	3	4	7
Meta	20	24	12	8	1	36
Network	33	39	40	3	2	8
Type	130	45	68	119	117	76
Linking	125	92	59	74	100	81
Global	21	11	13	0	0	0
Relation	21	14	11	33	16	20
Content	269	114	219	230	205	231
Operation	84	27	44	58	47	77
Total	710	378	473	528	492	536



Verbalization Segments

Category	Treatment T1	Treatment T2	Treatment T3	Control C1	Control C2	Control C3
Unknown	6	10	7	3	6	7
Planning	20	35	37	45	23	16
Intention	131	83	71	114	111	96
Evaluation	148	77	103	51	41	69
Attention	405	173	255	315	311	348
Total	710	378	473	528	492	536