Concept Maps as Hypermedia Components

Brian R. Gaines and Mildred L. G. Shaw
Knowledge Science Institute
University of Calgary
Alberta, Canada T2N 1N4
{gaines, mildred}@cpsc.ucalgary.ca

Abstract

Concept mapping has a history of use in many disciplines as a formal or semi-formal diagramming technique. Concept maps have an abstract structure as typed hypergraphs, and computer support for concept mapping can associate visual attributes with node types to provide an attractive and consistent appearance. Computer support can also provide interactive interfaces allowing arbitrary actions to be associated with nodes such as hypermedia links to other maps and documents. This article describes a general concept mapping system that is open architecture for integration with other systems, scriptable to support arbitrary interactions and computations, and cutomizable to emulate many styles of map. The system supports collaborative development of concept maps across local area and wide area networks, and integrates with World-Wide Web in both client helper and server gateway roles. A number of applications are illustrated ranging through education, artificial intelligence, active documents, hypermedia indexing and concurrent engineering. It is proposed that concept maps be regarded as basic components of any hypermedia system, complementing text and images with formal and semi-formal active diagrams.

1 Introduction

Hypertext systems commenced with simple facilities for creating nonlinear texts by embedding within a text links to other texts (McKnight, Dillon and Richardson, 1991). Hypermedia systems are a natural evolution of the linked content notion in which links can be embedded in any presentable item, and connect to any other one (Barrett, 1989). In particular, diagrams become hypermedia objects with the possibility of links from text into diagrams, and from within diagrams to other diagrams, text and general hypermedia objects. *Concept maps* are a form of diagram specifically targeted to provide *visual languages* similar in their characteristics to natural language text in that they can be subject to syntactic and semantic constraints, and their representational capacity can range from the fairly informal to the extremely formal.

Concept maps have been used in education, policy studies and the philosophy of science to provide a visual representation of knowledge structures and argument forms. They provide a complementary alternative to natural language as a means of communicating knowledge. In many disciplines various forms of concept map are already used as formal knowledge representation systems, for example: semantic networks in artificial intelligence, bond graphs in mechanical and electrical engineering, CPM and PERT charts in operations research, Petri nets in communications, and category graphs in mathematics.

Support of a general concept mapping component is appropriate in the architecture of any hypermedia system. The work reported in this article has originated from the development of systems to support large volumes of heterogeneous multimedia material generated in a variety of contexts such as knowledge acquisition (Gaines and Shaw, 1992b), sports coaching (Vickers and Gaines, 1988) and large-scale project support (Gaines and Norrie, 1994). The primary requirement has been to give end users access to multimedia material through indexing mechanisms that are simple and natural to use. A further requirement has been to support shared access to material through a variety of interfaces that reuse the same material for different purposes (Kremer and Gaines, 1994).

These requirements have been addressed by the development of a general visual language technology supporting customizable interactive concept maps (Gaines and Shaw, 1993c) and semantic networks (Gaines, 1991). The maps may be used as stand-alone documents or embedded as interactive pictures in active documents (Gaines and Shaw, 1993b). The technology is open architecture, and user interaction with the concept maps may be programmed to initiate any activity feasible on the host system. Thus maps may be linked to other maps for retrieval purposes, and may be used to retrieve, play and edit multimedia material either through the host system or under its control. The technology is designed to be shared and operate over local or wide area networks so that multiple users can access the same maps, and maps can be used to retrieve material from remote sites.

Concept maps may be used for the indexing and retrieval of hypermedia material, providing an attractive, meaningful and easy to use interface. In addition, maps with formal semantics may be used to provide a programming interface for the control of multimedia material. For example, Petrinets may be used to specify the essential synchronizations in the playback of a mix of multimedia material, and to index through the time relations in such material. Semantic networks may be used to represent knowledge bases that are operational and can be used for knowledge-based access to material.

The following sections give an overview of concept maps and their applications, formal semantics for concept maps, an abstract model of general concept maps, the implementation of a general concept mapping component for hypermedia systems, its user interface and computational capabilities, its application to hypermedia indexing, collaborative applications of linked maps, and integration of concepts map with World-Wide Web systems for wide-area collaboration.

2 Concept Maps

Many disciplines developed visual languages for ‘concept maps’, ‘cognitive maps’ and ‘argument forms.’ In education, Ausubel’s (1968) cognitive learning theory led Novak (1977) to develop a system of concept maps that has been widely applied in the evaluation of students’ learning in the school system (Novak and Gowin, 1984). Figure 1 shows a concept map from these studies with two types of nodes, *concepts* shown by ovals, and *instances* shown by rectangles. These are linked by arrows labeled with relations such as *needed by, made of, changes*, and so on. The conceptual structure developed encompasses some of the physics and biological roles of water. The student has developed the map within broad guidelines as to what are concepts and instances, and that they are to be linked by labeled directed arrows denoting relations.

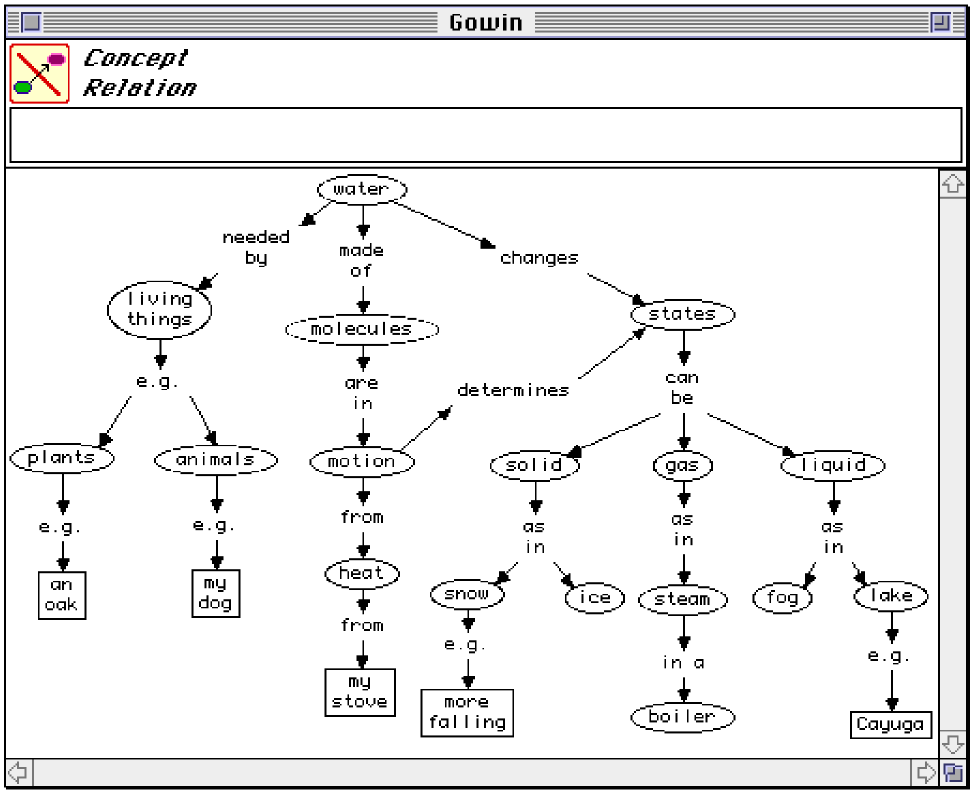


Figure 1 Concept map of student's knowledge (Novak and Gowin, 1984)

A wide variety of different forms of concept map have been applied in education (Lambiotte, Dansereau, Cross and Reynolds, 1989). They have also been used as tools to support the interviewing process in knowledge acquisition from experts, for example in the Wright-Patterson development of the pilot's associate (McNeese, Zaff, Peio, Snyder, Duncan and McFarren, 1990). In management, Axelrod (1976) proposed a form of concept maps as a means of representing the conceptual structures underlying decision making, and these have been used empirically to analyze organizational decision making (Eden, Jones and Sims, 1979), social systems (Banathy, 1991) and the policies of political leaders (Hart, 1977).

In artificial intelligence, Quillian (1968) developed a form of concept map that came to be termed *semantic networks* and used extensively for formal knowledge representation. In linguistics, Graesser and Clark (1985) have developed an analysis of argument forms in text in terms of structured concept maps with eight node types and four link types, and Woodward (1990) has developed tools to extract such maps from text. In the history of science, the dynamics of concept maps have been used to represent the processes of conceptual change in scientific revolutions (Nersessian, 1989; Thadgard, 1992).

In the philosophy of science, Toulmin (1958) developed a theory of scientific argument based on typed concept maps that is regarded as one of the major themes of the rhetoric of western thought (Golden, Berquist and Coleman, 1976). Figure 2 shows a concept map from these studies applied to the derivation that Harry, born in Bermuda, is a British citizen. The methodology prescribes various types of component that may be expected in argument forms such as *backing*, *warrant*, and so on, and these are indicated by textual labels rather than differing shapes.

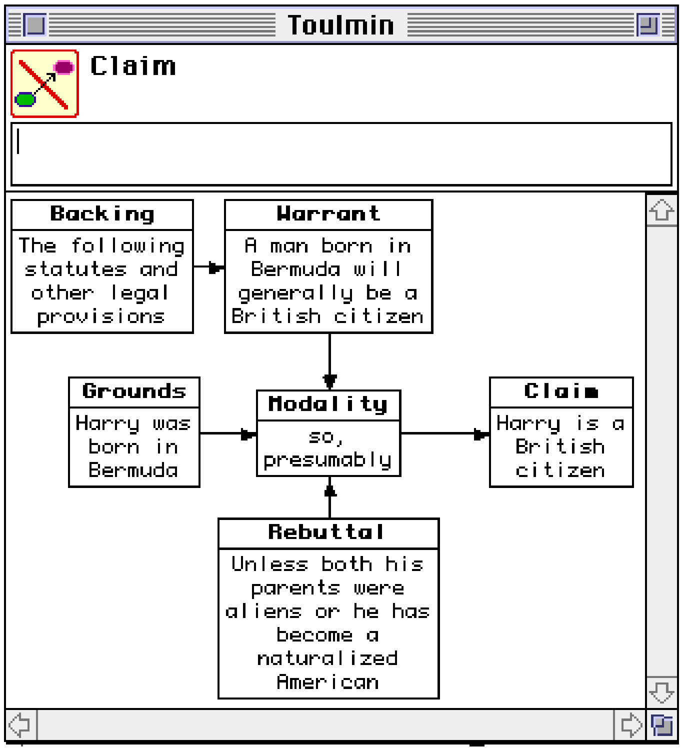
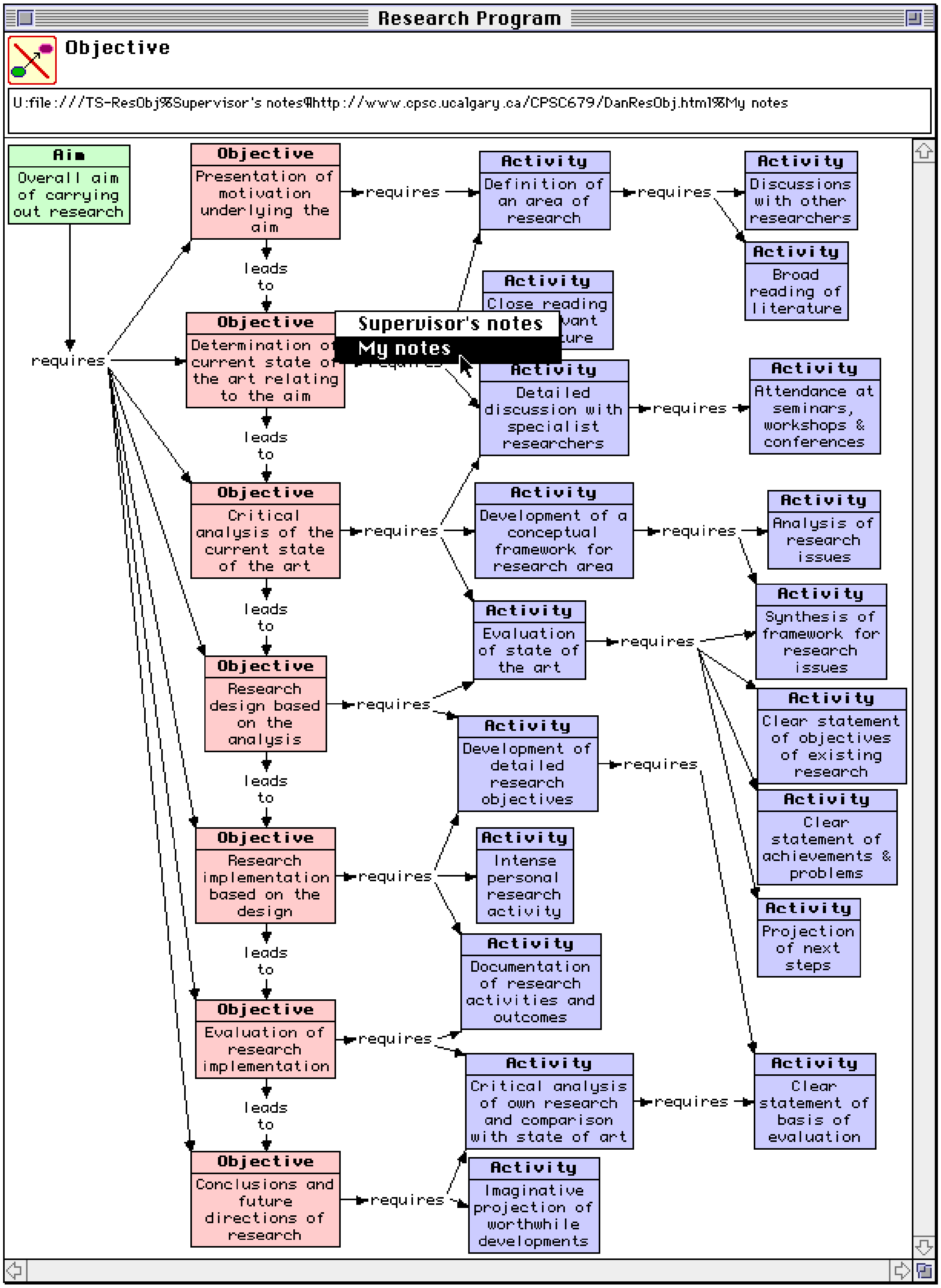

Toulmin
Claim
Backing
The following statutes and other legal provisions
Warrant
A man born in Bermuda will generally be a British citizen
Grounds
Harry was born in Bermuda
Modality
so, presumably
Claim
Harry is a British citizen
Rebuttal
Unless both his parents were aliens or he has become a naturalized American

Figure 2 Concept map of argument form (Toulmin, 1958)

Hypertext systems that display overviews of nodes and links provide a natural tool for the computer support of concept map development (Smolensky, Bell, Fox, King and Lewis, 1987), and specific tools to do so have been developed (Streitz, Hannemann and Thüring, 1989; Bernstein, 1992). Frames, semantic networks and hypertext structures have been compared (Travers, 1989), and Jonassen (1990) has shown how the elicitation of semantic networks may be used to structure hypertext. A variety of techniques and computer-based tools have also been developed for the analysis of concept maps. When they represent the strict semantics of a visual language for knowledge representation, it is appropriate to use deductive inference engines to draw logical conclusions from the maps (Sowa, 1991a). Even general concept maps can be analyzed in terms of their structure of their link relations (Cropper, Eden and Ackermann, 1990).

3 Hypermedia Concept Maps

Computer support of concept maps makes it simple to enhance their visual appearance through precise drawing and consistent use of color. It also allows the map to be used as an active computer interface providing links to, and control of, other materials. For example, Figure 3 shows an application in graduate education where students are introduced to the structure of a research program through an interactive concept map that may be edited and linked to their own research (Gaines and Shaw, 1993c). The node types and content may be edited by the students, as may a database of information attached to the nodes that provides links to other concept maps and files, either locally or across the Internet. In Figure 3, the student has moused over a node on objectives concerned with the determination of the current state of the art, and moused down when the cursor changed to a popup menu icon to access a menu giving access to a local file of her supervisor’s notes and a web file of her own notes put up as part of a course.


Research Program
Objective
Aim
Overall aim of carrying out research
requires
Objective
Presentation of motivation underlying the aim
leads to
Objective
Determination of current state of the art relating to the aim
leads to
Objective
Critical analysis of the current state of the art
leads to
Objective
Research design based on the analysis
leads to
Objective
Research implementation based on the design
leads to
Objective
Evaluation of research implementation
leads to
Objective
Conclusions and future directions of research
requires
Activity
Definition of an area of research
requires
Activity
Discussions with other researchers
Activity
Broad reading of literature
Activity
Close reading
vant
ure
Supervisor's notes
My notes
Activity
Detailed discussion with specialist researchers
requires
Activity
Attendance at seminars, workshops & conferences
requires
Activity
Development of a conceptual framework for research area
requires
Activity
Analysis of research issues
Activity
Evaluation of state of the art
requires
Activity
Synthesis of framework for research issues
requires
Activity
Development of detailed research objectives
requires
Activity
Clear statement of objectives of existing research
Activity
Clear statement of achievements & problems
Activity
Projection of next steps
requires
Activity
Intense personal research activity
Activity
Documentation of research activities and outcomes
requires
Activity
Critical analysis of own research and comparison with state of art
requires
Activity
Clear statement of basis of evaluation
requires
Activity
Imaginative projection of worthwhile developments

U:file:///TS-ResObj%Supervisor's notes¶http://www.cpsc.ucalgary.ca/CPSC679/DanResObj.html%My notes

Figure 3 Concept map on carrying out graduate research (Gaines and Shaw, 1993c)

Computer support also makes it possible to manage very large concept maps that are difficult to manage in paper form. The upper window of Figure 4 shows part of a map with some 900 nodes that was originally developed on large white boards from US Air Force pilots as part of the knowledge acquisition process for the design of the pilot's associate expert system (McNeese et al., 1990). It took 6 person-hours to enter this entire map from the printouts in the report cited. An auxiliary helper window at the bottom shows the whole map in miniature with a rectangle at the lower right scrolling with the upper window to indicate the location in the total map of the part being viewed and edited. A "fit-to-screen" command may be used to display the entire map in the editing window also so that large parts of it may be moved or copied and pasted in one operation. In general, such large maps are usually partitioned into small modules that are linked through hypermedia links such as those as illustrated in Figure 3. However, if required, large maps are readily managed through a computer support system.

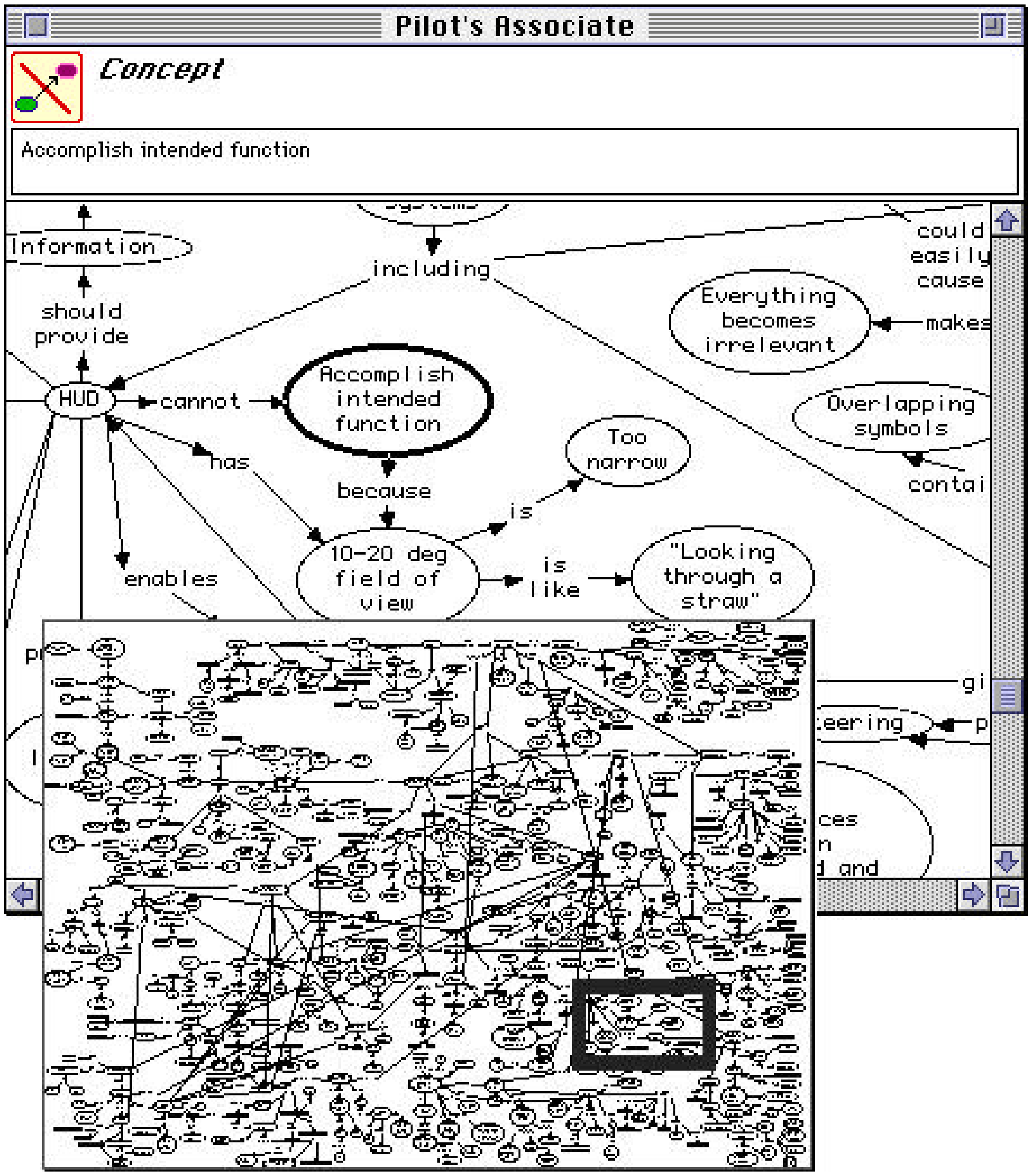


Figure 4 900 node concept map for design of pilot's associate (McNeese et al., 1990)

4 Formal Concept Maps

The concept maps of Figures 1 through 4 already involve more formal constraints than do diagrams in general. Novak and Gowin's methodology involves what are essentially syntactic constraints on the form of a concept map. Toulmin's methodology involves semantic constraints on what constitutes a legitimate argument form. However, neither methodology is sufficiently formal, nor intended to be so, that the map can be translated directly into a computational expression.

Many disciplines have developed diagramming techniques that constitute formal visual languages representing operational knowledge in diagrammatic form, for example, CPM and PERT charts (Moder and Phillips, 1970), Petri nets (Reisig, 1985), bond graphs (Karnopp and Rosenberg, 1975), and category-theoretic diagrammatic proofs (Mac Lane, 1971). For example, Figure 5 shows a bond graph for the components of a bicycle from an engineering design system (Gui and Mäntylä, 1993). Bond graphs are used in a wide range of engineering disciplines and have formal computational methodologies associated with their application (Karnopp, Rosenberg and van Dixhorn, 1989).

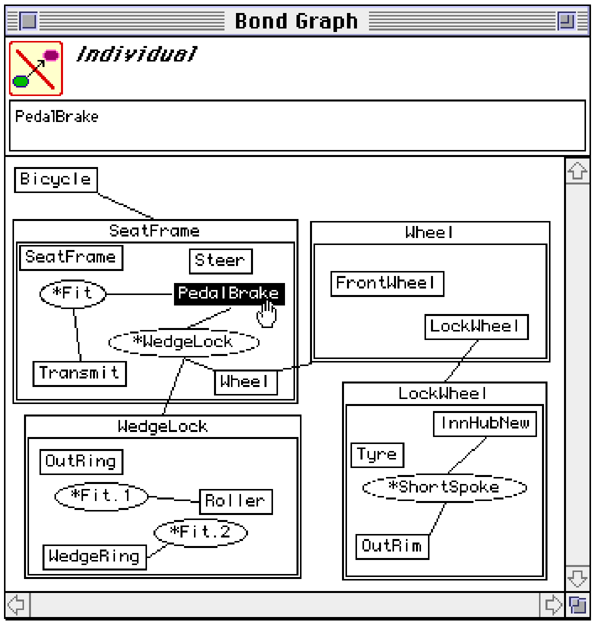


Figure 5 A bond graph in engineering design (Gui and Mäntylä, 1993)

The philosopher and logician Charles Sanders Pierce developed his *existential graphs* as a formal reasoning technique for logical inference (Roberts, 1973), and in recent years there has been growing interest in formal status of visual proofs in mathematics (Barwise and Etchemendy, 1990; Shin, 1994). In artificial intelligence, Sowa (1984) has developed Peirce's graphs as formal *conceptual structures* for the representation of logical inference from natural language statements.

Figure 6 shows a conceptual graph from an article by Sowa (1991b) embedded as active hypermedia object within the article using an *active document* technology that allows the map to be edited within the document processor and interrogated by other programs as a formal knowledge base (Gaines and Shaw, 1993b). The graph represents the English language statement that "Tom believes that Mary wants to marry a sailor." It may be translated automatically into the linear, textual form of conceptual graphs as shown beneath it, and both diagram and textual form may be translated into predicate calculus. The textual form can also be loaded into the PEIRCE (Ellis and Levinson, 1992; Ellis, Levinson and Robinson, 1994) inference engine for conceptual graphs, and queried as a deductive database.

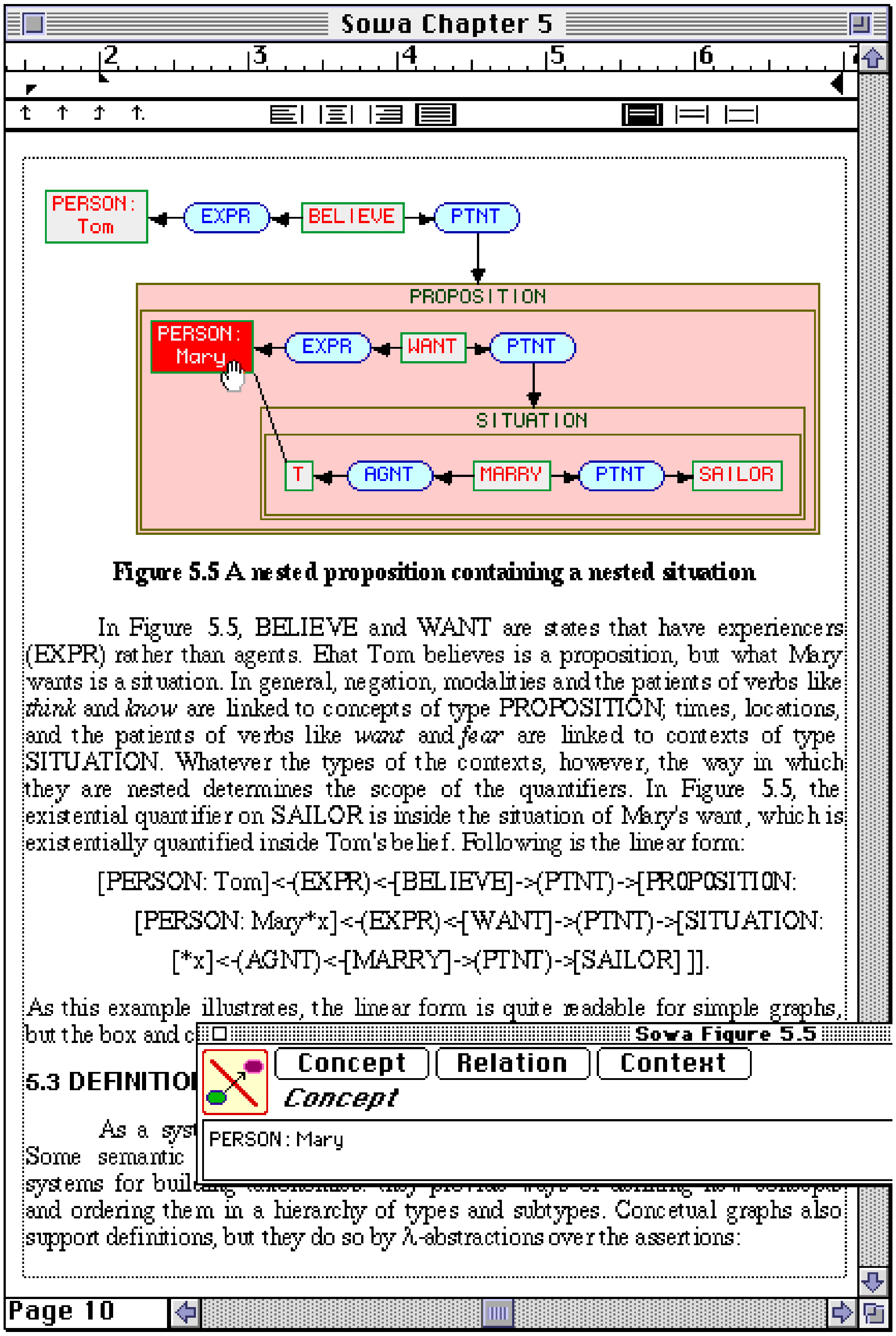


Figure 5.5 A nested proposition containing a nested situation

In Figure 5.5, BELIEVE and WANT are states that have experiencers (EXPR) rather than agents. Ehat Tom believes is a proposition, but what Mary wants is a situation. In general, negation, modalities and the patients of verbs like *think* and *know* are linked to concepts of type PROPOSITION; times, locations, and the patients of verbs like *want* and *fear* are linked to contexts of type SITUATION. Whatever the types of the contexts, however, the way in which they are nested determines the scope of the quantifiers. In Figure 5.5, the existential quantifier on SAILOR is inside the situation of Mary's want, which is existentially quantified inside Tom's belief. Following is the linear form:

[PERSON: Tom]<-(EXPR)<-[BELIEVE]->(PTNT)->[PROPOSITION:
[PERSON: Mary*x]<-(EXPR)<-[WANT]->(PTNT)->[SITUATION:
[*x]<-(AGNT)<-[MARRY]->(PTNT)->[SAILOR]]].

As this example illustrates, the linear form is quite readable for simple graphs, but the box and c

5.3 DEFINITIO

As a syst
Some semantic
systems for buil
and ordering them in a hierarchy of types and subtypes. Concetual graphs also support definitions, but they do so by λ-abstractions over the assertions:

PERSON: Mary

Figure 6 Concept map after a chapter by Sowa (1991b) embedded in an active document

The active document technology has been used to write “knowledge-based” papers (Gaines and Shaw, 1992a) in which concept maps representing formal knowledge structures are embedded in the document as active, editable entities having the user interface facilities of the same maps displayed in their own windows. To allow the document to print normally, the editing facilities at the top of the windows shown in the previous figures are normally hidden. They appear in a floating dialog box as shown near the bottom right of Figure 6 when the user double clicks in the concept map. The electronic version of the document is active, providing typographic text and page layout facilities, versioning, hypermedia sound and movies, hypertext links, and knowledge structures represented in a visual language. It can be read as a hypermedia document and also interrogated as a knowledge-based system for problem-solving. The paper version of the document is produced by printing the electronic version. It loses its active functionality but continues to act as a record of the knowledge in the document. Concept maps embedded in digital documents as active diagrams allow, for example, a conference proceedings to be issued as a computational document that makes the systems described in the proceedings operationally available.

5 Semantic Networks

Visual presentation of knowledge structures has been an attractive feature of semantic networks since their inception (Quillian, 1968). In knowledge acquisition, in particular, the presentation of formal knowledge to those from whom it has been elicited is important to its validation (Gaines and Shaw, 1992b). There are many techniques for such acquisition but they ultimately result in an operational knowledge base with formal semantics. However, the expression of this knowledge base in the formal language used by the system is usually not very comprehensible to non-programmers (Nosek and Roth, 1990). A visual language that is both comprehensible and formal offers attractive possibilities not only for comprehension but also for editing, and for parts of the knowledge acquisition process itself.

The early development of semantic nets resulted in criticisms that the semantics of particular diagrams was not well-defined (Woods, 1975; Brachman, 1977). Nodes, arcs and their labels could be used very freely and ambiguously and diagrams were subject to differing interpretations. In the 1970s there were proposals for network formalisms with well-defined semantics (Cercone and Schubert, 1975; Brachman, 1979; Fahlman, 1979), and this became feasible as formal semantics were developed for terminological knowledge representation systems (Borgida, Brachman, McGuiness and Resnick, 1989; Baader and Hollunder, 1991; Borgida and Patel-Shneider, 1994). A formal semantics for concept maps for terminological logic systems has been specified (Gaines, 1991), a major example of its application in the solution of a room allocation system has been given (Gaines, 1994b), and the relationship between the maps and the knowledge representation data structures has been analyzed in detail (Gaines, 1994a).

Figure 7 shows a semantic network representing some knowledge about software copyright disputes (Madan, 1994) represented for a terminological logic system. At the top left is a general definition of a primitive concept “legal dispute.” Immediately below it the concept “copyright dispute” is defined as a “legal dispute” whose “claim” role includes “copyright violation.” Below this the concept “software copyright dispute” is defined as a “copyright dispute” whose “domain” role includes “software.” Below this the concept “user interface copyright dispute” is defined as a “software copyright dispute” whose “basis” role includes “user interface similarity.” This concept is the premise of the rule “ui copyright cases,” and when an individual satisfying this premise is recognized then the rule fires and the conclusion is asserted that the individual has its “cases for plaintiff” role filled by “Cross vs Mirror” and “PrintShop vs PrintMaster,” and its “cases against plaintiff” role filled by “Apple vs Microsoft.”

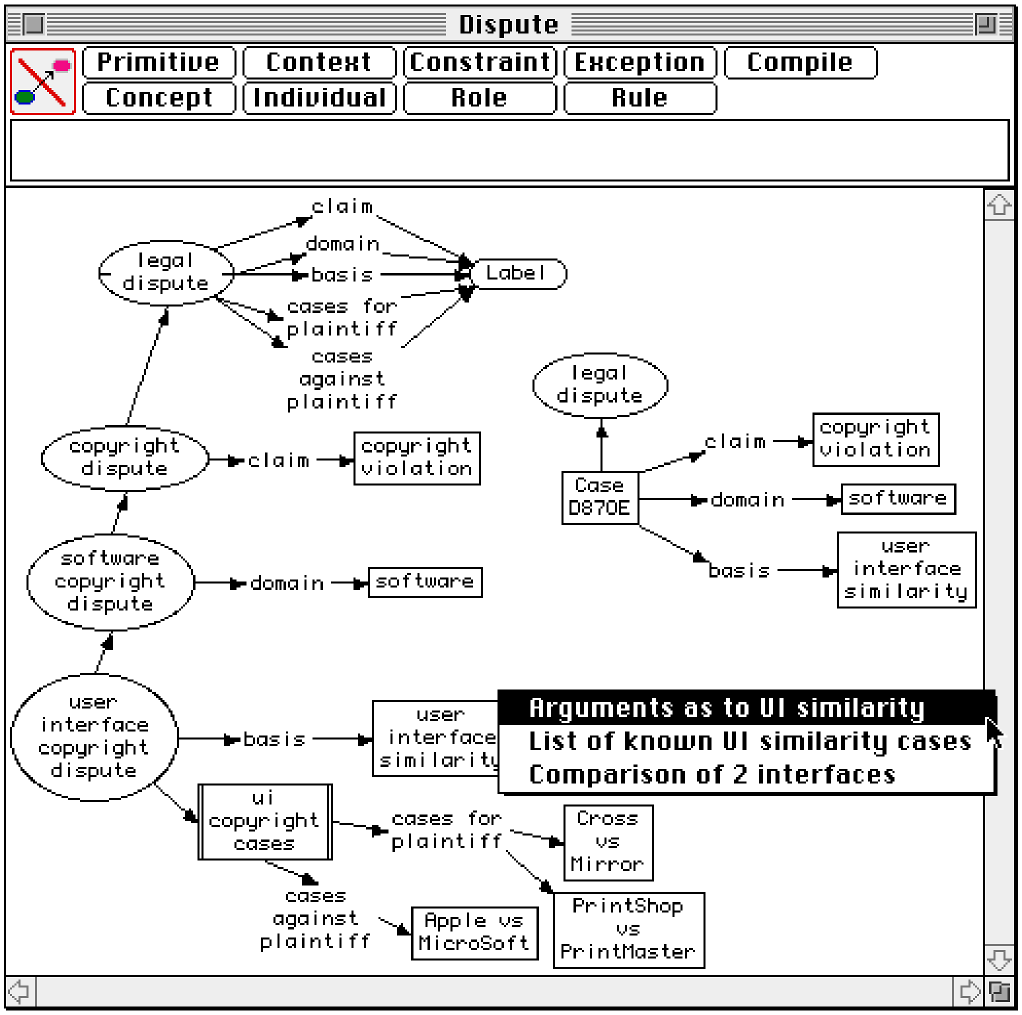


Figure 7 Concept map for a semantic network for terminological knowledge representation

On the right in Figure 7, the individual “Case D87OE” is asserted to be an instance of a “legal dispute” whose “claim” role is filled by “copyright violation,” “domain” role by “software,” and “basis” role by “user interface similarity.”

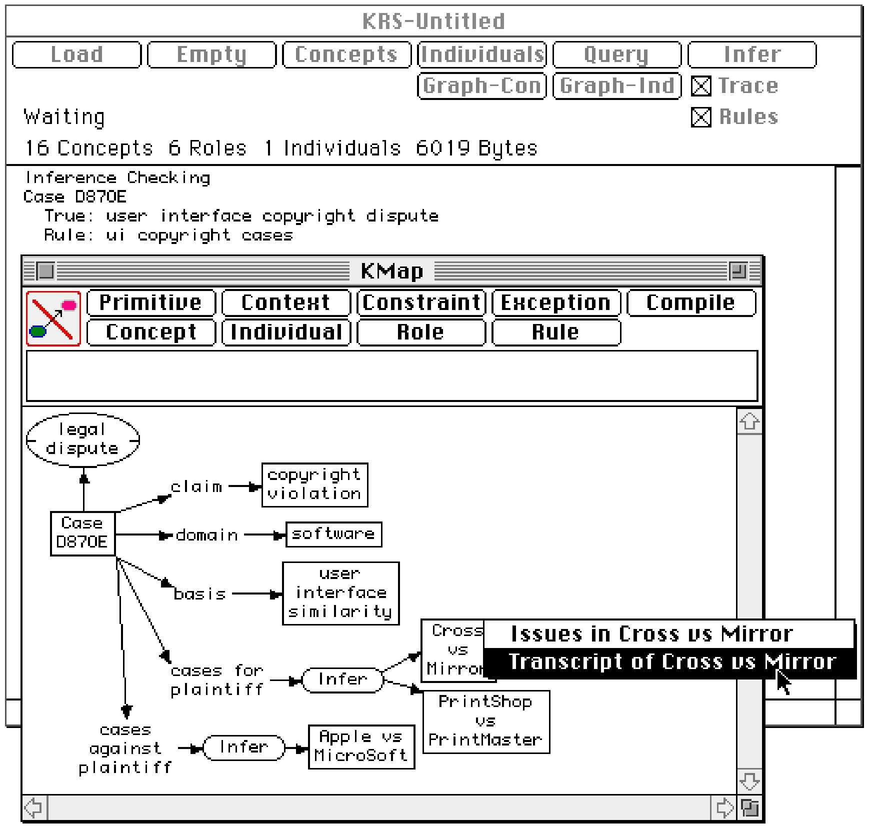
When the “Compile” button is clicked the semantic network is converted to an equivalent textual form as shown in Figure 8.

```
Primitive(legal dispute
  (All claim (Label))
  (All domain (Label))
  (All basis (Label))
  (All cases for plaintiff (Label))
  (All cases against plaintiff (Label))
)
Concept(copyright dispute, legal dispute
  (All claim (Include copyright violation))
)
Concept(software copyright dispute, copyright dispute
  (All domain (Include software))
)
Concept(user interface copyright dispute, software copyright dispute
  (All basis (Include user interface similarity))
  (Rule ui copyright cases)
)
Rule(ui copyright cases
  (All cases for plaintiff (Include Cross vs Mirror, PrintShop vs PrintMaster))
  (All cases against plaintiff (Include Apple vs MicroSoft))
)
Individual(Case D87OE, legal dispute
  (Fills claim, copyright violation)
  (Fills domain, software)
  (Fills basis, user interface similarity)
)
```

Figure 8 Textual form of knowledge structure compiled from Figure 7

Figure 9 shows inference taking place when the knowledge structure of Figure 8 is loaded into a KRS knowledge representation server, or that of Figure 7 is “pasted” into the server window and automatically compiled. The trace in the upper part shows the inferences being made. Clicking on the button “Graph-Ind” causes the results of inferences about individuals to be graphed as a concept map as shown in the lower part. It can be seen that “Case D87OE” has been inferred to have its “cases for plaintiff” role filled by “Cross vs Mirror” and “PrintShop vs “PrintMaster,” and its “cases against plaintiff” role filled by “Apple vs Microsoft.”

As in previous examples, the popup menus in Figures 7 and 9 provide links to a database of associated information, in this example a database of legal background material, case summaries and transcripts. This enables the nodes that have been added through inference to be used to retrieve further information that has been inferred to be relevant to the case described. More generally, this can be seen as a technique whereby deductive inference can been used to create new links between hypermedia structures.



```
Waiting
16 Concepts  6 Roles  1 Individuals  6019 Bytes
```

```
Inference Checking
Case D870E
  True: user interface copyright dispute
  Rule: ui copyright cases
```

Figure 9 Inference from knowledge structure of Figures 7 and 8 output as concept map

6 What is a Concept Map?

It may seem a rhetorical device to leave the definition of a concept map to this late stage in the article. However, it should be clear from the many example given, and the many more to be found in the literature cited, that the term "concept map" is used to encompass a wide range of diagrammatic knowledge representations. As in Wittgenstein's (1953) analysis of the problem of developing a clear, unambiguous and complete definition of the commonly understood term, "game," neither the term "concept" nor the term "map" have precise definitions. Each term is commonly used colloquially, and both have a variety of different precisifications as technical terms in different disciplines. The compound term "concept map" inherits connotations from these varying usages, and has usages of its own. Hence, it has appropriate to exemplify the term in a range of applications before addressing its definition.

In explicating, or deconstructing, the notion of concept map based on the examples in the literature, one finds a need for three levels of analysis: from an *abstract* perspective concept maps as nodes linked by arcs may be seen as representations of *graphs*, using the term as defined in mathematics (Berge, 1958); from a *visualization* perspective concept maps may be seen as *diagrams*, using the term to mean a drawing with reasonably well-understood semiotics for some community (Bertin, 1983); from a *discourse* perspective concept maps may seen as a way of representing and communicating knowledge through *visual languages* (Chang, Ichikawa and Ligomenides, 1986; Saint-Martin, 1990). Each perspective draws out different common threads from the many exemplars of concept maps in the literature.

From an abstract perspective, the basic concept map data structure is a *sorted hypergraph* consisting of *typed nodes* some of which are linked. Each node has a *type*, a *unique identifier* and a *content* (which may itself be structured, for example, as label plus other data). A node may *enclose* other nodes giving the graph a *hypergraph* structure (Berge, 1973) in which a single link may connect sets of nodes. Links may be *directed* or *undirected*, represented visually by lines between nodes with, or without, arrow heads. Links may themselves be typed in some applications.

From a visualization perspective, to provide a consistent relation between the visual features as signs and their semiotic infrastructure, the *visual attributes* of nodes and links need to be in one-to-one unique correspondence with their types. The node type may determine, and itself be determined by, the node shape, frame color, fill color, whether the type name is displayed and, if so, in what type face, style, size and color, and whether part of the content is displayed and, if so, in what type face, style, size and color. If the links are types, their types may determine, and be determined by, line thickness, color, cross-hatching, or other forms of decoration. Visual attributes that are not part of this determination may be used for other purposes, such as emphasis, but it is important that such usage does not obscure the primary relation.

From a discourse perspective, the abstract structure represented as a diagram in visual terms consistent with it has meaning as knowledge representation and communication because it is subject to *interpretation* by some *reference community*. In this there is an exact parallel between natural language and visual language—the abstract grammatical structures and their expressions in a medium take on meaning only through the practices of a community of discourse. Some communities may find it appropriate to use language in a loose, associative fashion, whereas others may wish to use it with technical precision, and usage may well be mixed with informal and formal sub-languages combined in actual discourse.

These three perspectives provide the framework for the development of tools to support concept mapping activities, and were the basis of the design of KMap as an open architecture concept mapping tool used to produce the maps shown in this article.

7 Concept Map Structure and Implementation

KMap was written as a module in a C++ class library that provides a wide range of data structures, communication protocols, scripting, graphic user interfaces, document processing, hypermedia, knowledge representation and inference (Gaines, 1994a). This makes it possible to support a wide range of applications of concept maps, including embedding in documents, information retrieval, and collaborative access over local area and wide area networks.

The KMap implementation has three layers corresponding to those discussed in the previous section. From the abstract perspective, a hypergraph representation data class has been implemented which supports typed nodes, and this is sub-classed to support typed links between nodes. From the visualization perspective, the node and link types have associated visual representation classes that record the parameters determining the appearance of the types and the methods for drawing them. From the discourse perspective, translators are supplied for some communities that recode the abstract data into an alternative form of knowledge representation, and, in general the graph structure can be exported to other tools that have such translation facilities.

This linking of appearance to node type allows the abstract sorted graph structure to be presented clearly to the user through the interface. The availability of a rich variety of visual presentation facilities is what enables KMap to be set up to emulate a wide range of existing forms of concept map. For example, in Figure 1 there are 3 types of node indicated by the presence and shape of the frame, and the type name itself is not displayed. In Figure 2 there are 6 types of nodes, all with the same shape but differentiated by the type name as a heading. In Figure 5 the form of the 3 node types was chosen to correspond to bond graph conventions; in Figure 6 the 3 node types correspond to conceptual graph conventions; and so on. Note that what may appear to be link labels are treated as nodes in KMap. This allows more concise and meaningful maps to be drawn since the "link label" node can have multiple incoming and outgoing arrows.

The content of a node is split into a visible component and an invisible component. The user tends to think of the visible component as the node name or content. The invisible component is used for computational purposes such as hypermedia links and information retrieval. Interactivity, such as the availability of popup menus, is determined by the node type, but the content of such interaction, such as the items in popup menus, is determined by the invisible component of the node content. Because the type and content of a node may be edited by the user, for example as part of a process of knowledge refinement, a unique identifier is associated with each node that may be used to preserve links between it and other data structures.

In most applications the node types are set up in advance, and the end user of a concept map has available a preset system of node types. However, the node type editor is available and the types and their appearance can be edited during ongoing development of a concept map if required. Figure 10 shows a concept map in a multimedia information retrieval system with the user selecting the type "Movie" on a popup menu prior to adding a new node. The visible content is entered by typing it into the text editing box under this menu. The invisible content is entered by double clicking on the node to edit it while holding down the "option" key on the Macintosh. KMap normally automatically word wraps the visible content to form a horizontal rectangle but this can be overridden by the user inserting the paragraph marker symbol, ¶, to indicate line breaks.

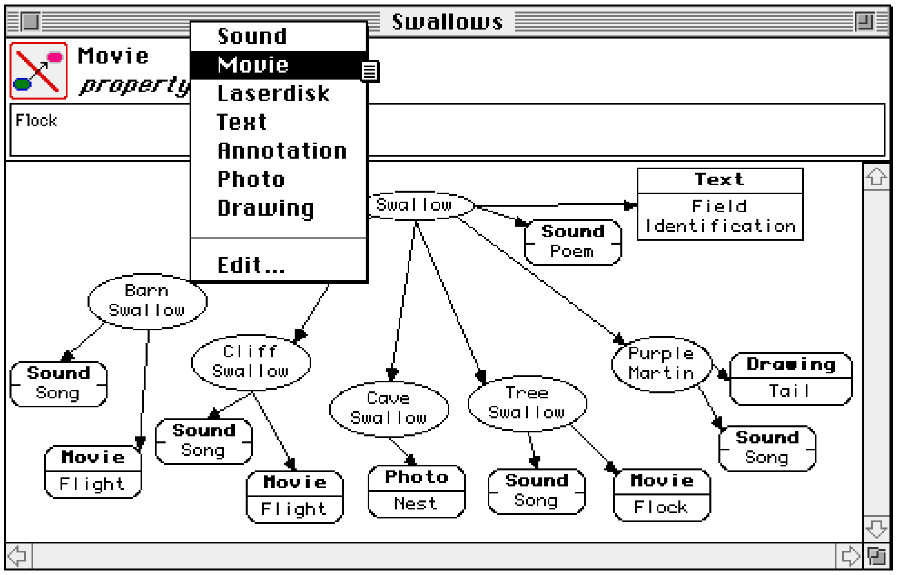


Figure 10 Concept map accessing multimedia materials

At the bottom of the popup menu the option "Edit…" is offered, and selecting this causes a node type editing window to open as shown in Figure 11. Near the top of its window, the editor provides a palette of shapes which can be associated with types, and in the bottom of the window all the types so far defined are shown. The shape and colors of existing types may be redefined and, when the "Apply" button is clicked, the new appearance will show up in the concept map in the window below.

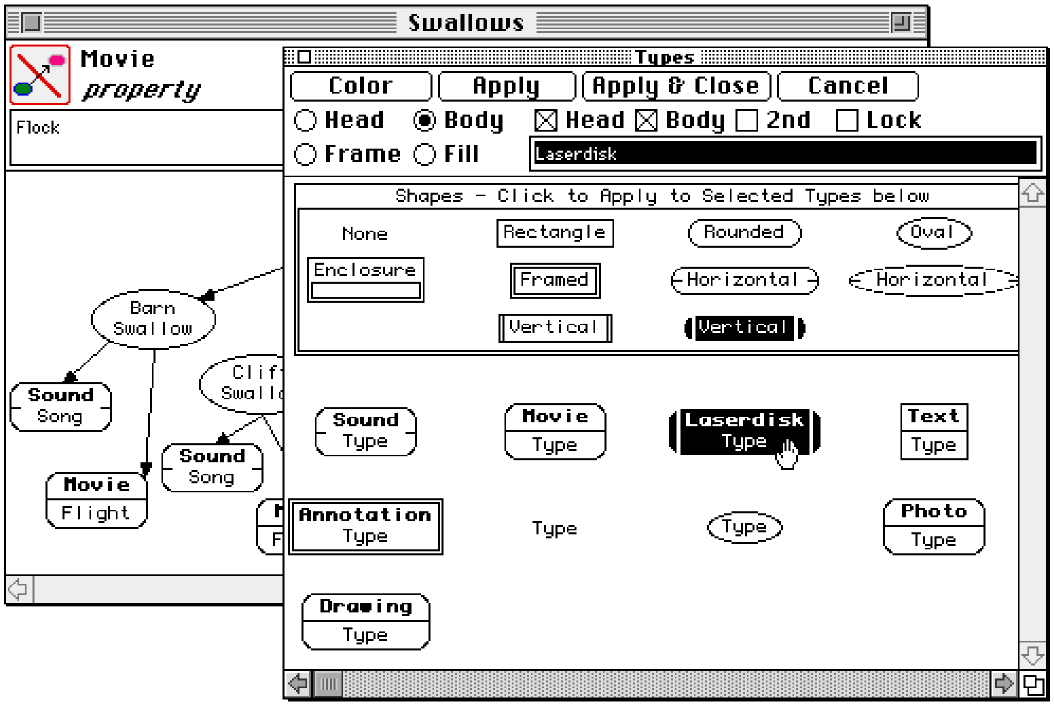


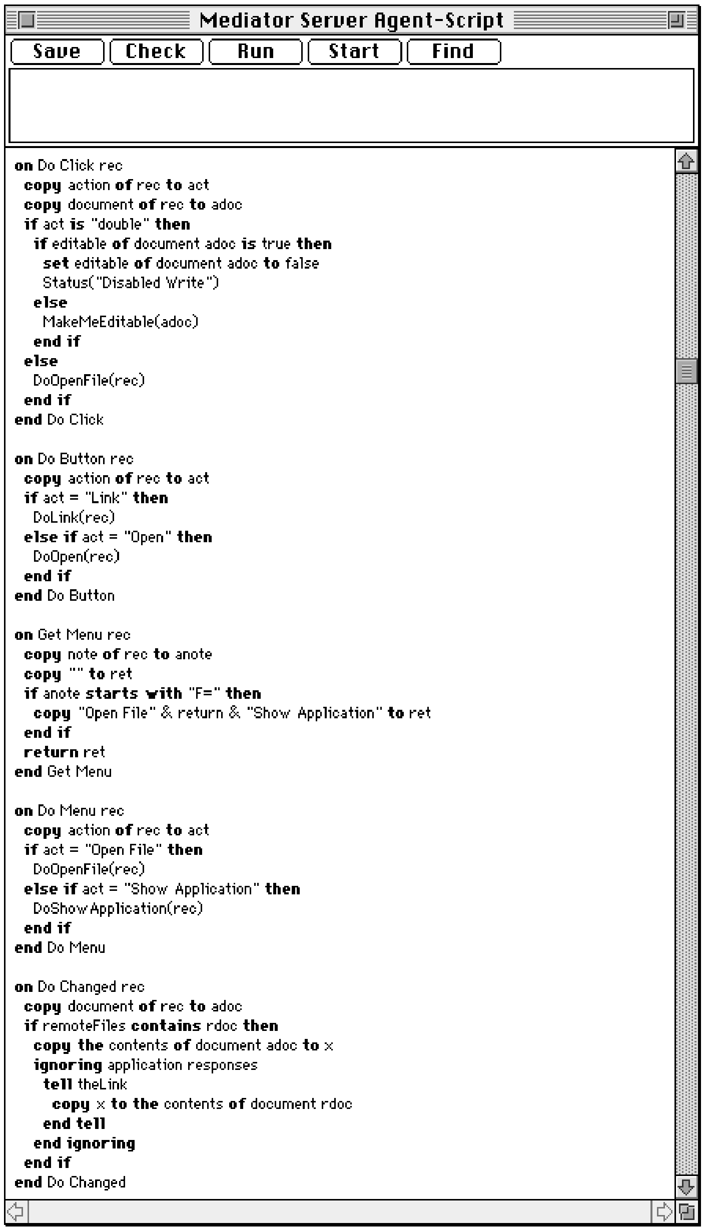
Figure 11 Entering concept map types to define a visual language

The user interface through which the types of nodes may be selected is itself customizable. In Figure 10 types are accessed through the two popup menus at the top left. Two menus are provided because node types often divide naturally into two types, for example a "node type" and a "link type" with the constraint that nodes whose type is in one class may be connected by arcs only to nodes whose type in the other. The concept maps in Figures 1, 3 and 6 have such a bipartite structure.

However, the menus are only defined in a Macintosh operating system "resource" whose number is stored with the concept map, and a wide variety of other user interfaces may be defined through alternative resources. For example, the "buttons" at the top of Figure 7 are an alternative to a popup menu. If the KMap software receives a message that a button has been clicked it checks whether the label of the button is a defined type name and if so, treats a button click in the same way as a popup menu selection. In addition, by setting a switch stored with a concept map the entire editing pane shown at the top of most of the concept map examples may be defined to be a floating palette that does not appear until a node is double clicked. This is an important capability when a concept map is embedded in a document as shown in Figure 6, and should have the appearance of a simple diagram when viewed or printed, but also act as an active interface if required. The appearance of the concept map editing pane as a floating dialog is shown near the bottom of Figure 6.

8 Concept Map Scripting

KMap is itself programmable through the Apple high-level object event protocol (Apple, 1993a), and can be driven by any of the scripting languages in Apple’s open scripting architecture such as AppleScript (Goodman, 1993), Frontier (Winer, 1992) and TCL (Ousterhout, 1994). Each concept map can have its own script which receives messages triggered by user interaction with the concept map. This enables KMap to be integrated with other applications, and user interaction with graphical structures in the visual language to be used to control any activity supported on the host computer or network. The development of a specific system involves writing scripts to provide the required functionality by drawing upon existing applications. Figure 12 shows part of the script associated with a server agent in the Mediator (Gaines and Norrie, 1994) collaborative system opened for editing.



```
on Do Click rec
 copy action of rec to act
 copy document of rec to adoc
 if act is "double" then
  if editable of document adoc is true then
   set editable of document adoc to false
   Status("Disabled Write")
  else
   MakeMeEditable(adoc)
  end if
 else
  DoOpenFile(rec)
 end if
end Do Click

on Do Button rec
 copy action of rec to act
 if act = "Link" then
  DoLink(rec)
 else if act = "Open" then
  DoOpen(rec)
 end if
end Do Button

on Get Menu rec
 copy note of rec to anote
 copy "" to ret
 if anote starts with "F=" then
  copy "Open File" & return & "Show Application" to ret
 end if
 return ret
end Get Menu

on Do Menu rec
 copy action of rec to act
 if act = "Open File" then
  DoOpenFile(rec)
 else if act = "Show Application" then
  DoShowApplication(rec)
 end if
end Do Menu

on Do Changed rec
 copy document of rec to adoc
 if remoteFiles contains rdoc then
  copy the contents of document adoc to x
  ignoring application responses
   tell theLink
    copy x to the contents of document rdoc
   end tell
  end ignoring
 end if
end Do Changed
```

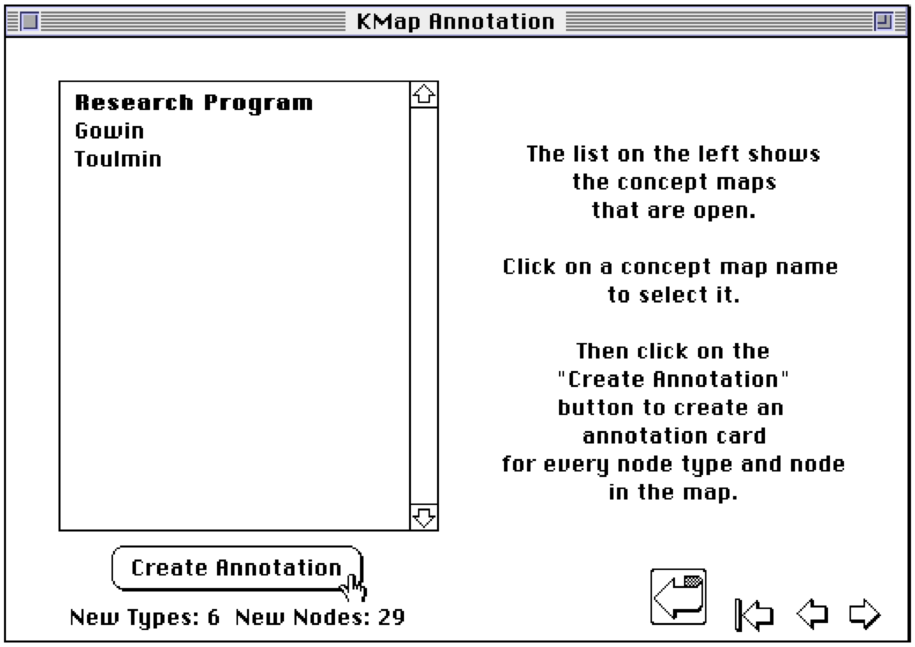
Figure 12 Script editor with part of script for major KMap events

The message handler routines shown are those activated by a mouse click in the concept map, by a button click in the associated dialog, by a mouse click that activates a popup menu, by a popup menu selection, and by any form of change in the concept map. Each message passes a record with associated parameters, such as the document name, whether the click was single or double, what node, if any, the click was in, and so on. The “Get Menu” handler that is activated by a mouse click when the cursor is a popup menu symbol returns the menu items that should be displayed so that the actual menu can be constructed dynamically based on the content of the node. The “Do Changed” handler in this example sends information about the changes in a concept map to update the same map open in KMap running on another machine, thus supporting collaborative editing of concept maps. Messages are sent to the script when a concept map is initially opened and when it is closed, supporting start up and completion actions. Messages sent from external applications may be routed to scripts in concept maps, supporting cross-application, cross-platform and cross-network integration.

It is often inappropriate to have a separate script for every concept map. The functionality required is usually common to a family of maps, and hence KMap supports shared scripts in which the script in one concept map acts as an “agent” serving a set of linked maps. Scripts may link together specific maps, or a script may set a flag defining itself as a default agent receiving messages from all maps that do not have relevant handlers in their own scripts or linked scripts.

9 Linking Concept Maps to other Knowledge Elicitation Technologies

The programmability of concept maps in KMap and the access of the script language to the full range of operating system capabilities makes it possible to integrate concept maps with other applications to develop tightly coupled systems that operate seamlessly to the end user. For example, Apple’s HyperCard is a useful adjunct to KMap in provide textual and multimedia annotation facilities. Figure 13 shows a HyperCard stack that links to KMap in such a way that the two systems keep one another updated about changes in relevant data structures.



The list on the left shows the concept maps that are open.

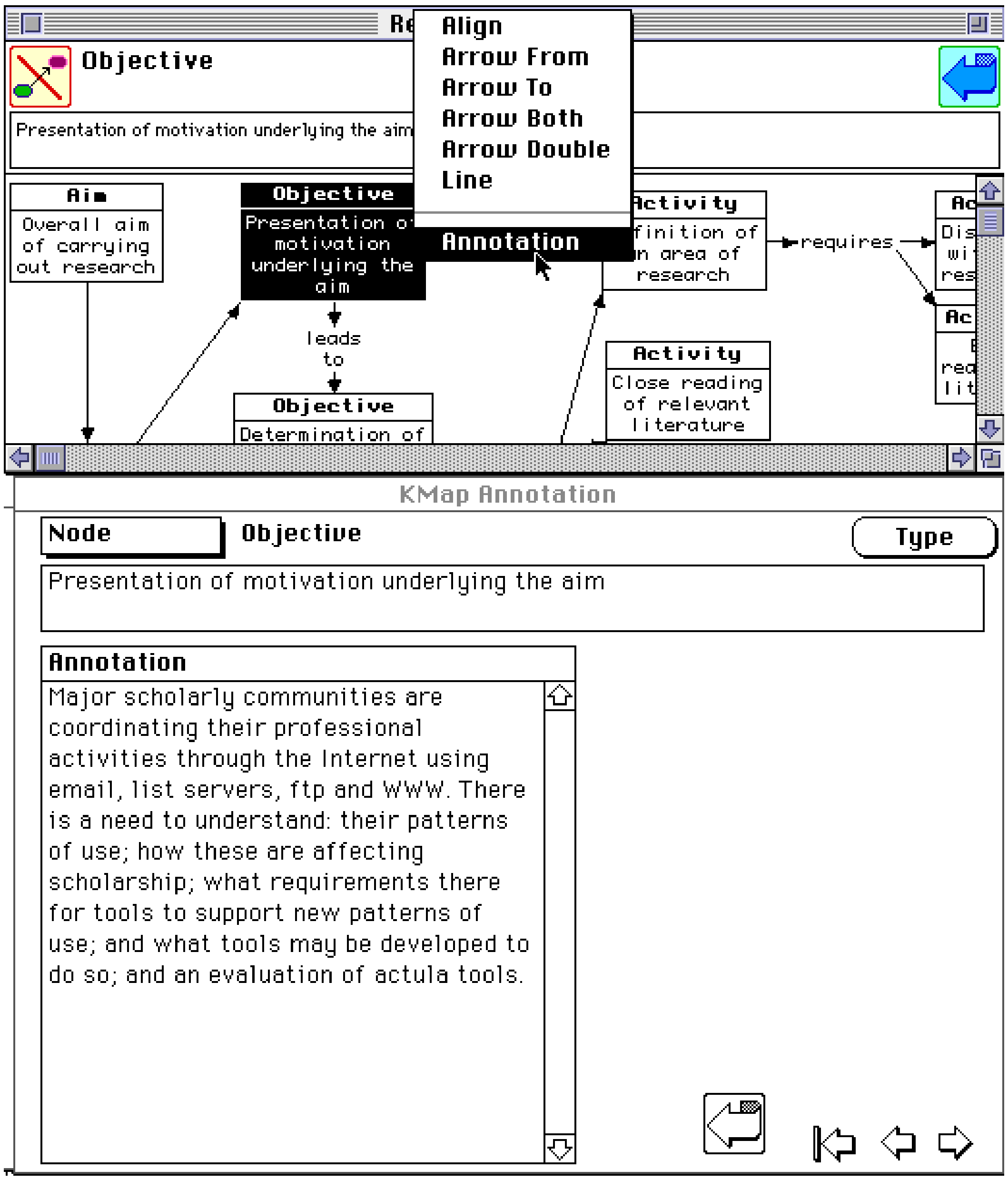
Click on a concept map name to select it.

Then click on the "Create Annotation" button to create an annotation card for every node type and node in the map.

New Types: 6 New Nodes: 29

Figure 13 Concept map annotation in HyperCard

The card shown in Figure 13 maintains a list of concept maps open in KMap—those of Figures 1 through 3. Selecting one of these in the list and clicking on the “Create Annotation” button causes HyperCard to create a preformatted annotation card for every type and every node in the concept map. Figure 14 shows a popup menu being activated for one node in the graduate research concept map of Figure 3. The access to HyperCard has given the menu an extra option at the bottom, “Annotation”, and selecting this causes the HyperCard stack to come to the front displaying the annotation card for that node. If a card for the node did not already exist, one would be created thus allowing annotation for a new node to be entered very simply. The student may enter textual annotation into a scolling box on the left, and populate the rest of the card with other relevant material, such as images and buttons linking to other cards.


Objective
Presentation of motivation underlying the aim
Align
Arrow From
Arrow To
Arrow Both
Arrow Double
Line
Annotation
Aim
Overall aim of carrying out research
Objective
Presentation of motivation underlying the aim
leads to
Objective
Determination of
Activity
finition of n area of research
requires
Activity
Close reading of relevant literature
KMap Annotation
Node
Objective
Type
Annotation

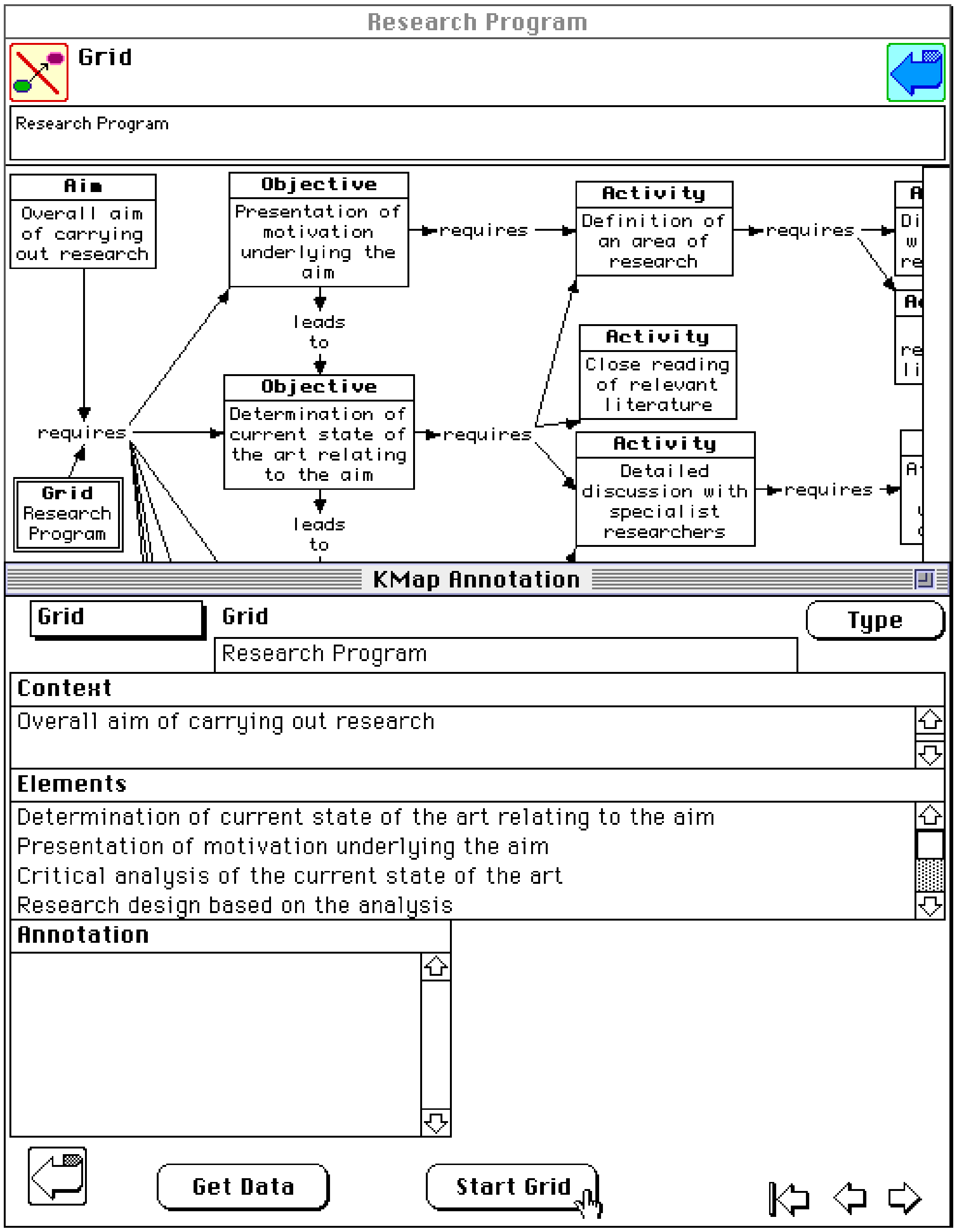
Presentation of motivation underlying the aim

Major scholarly communities are coordinating their professional activities through the Internet using email, list servers, ftp and WWW. There is a need to understand: their patterns of use; how these are affecting scholarship; what requirements there for tools to support new patterns of use; and what tools may be developed to do so; and an evaluation of actula tools.

Figure 14 Accessing the annotation for a concept

The “Type” button at the top right of the card links to the annotation card for the node type. This will often be set up in advance as a description of how that type of node is intended to be used in this form of concept map.

More complex forms of integration are readily created as extensions to the HyperCard stack. For example, it is common for lists of related entities to be generated in concept maps that are well-suited to more structured knowledge elicitation using repertory grid techniques (Bradshaw, Ford, Adams-Webber and Boose, 1993; Gaines and Shaw, 1993a) from personal construct psychology (Kelly, 1955). Figure 15 shows the annotation stack being used to generate automatically an initial set of elements to be used in a grid elicitation tool.


Research Program
Grid
Research Program
Aim
Overall aim of carrying out research
Objective
Presentation of motivation underlying the aim
requires
Activity
Definition of an area of research
requires
leads to
Objective
Determination of current state of the art relating to the aim
requires
Activity
Close reading of relevant literature
Activity
Detailed discussion with specialist researchers
requires
requires
Grid
Research Program
leads to
KMap Annotation
Type
Get Data
Start Grid

Grid | Grid
Research Program

Context
Overall aim of carrying out research

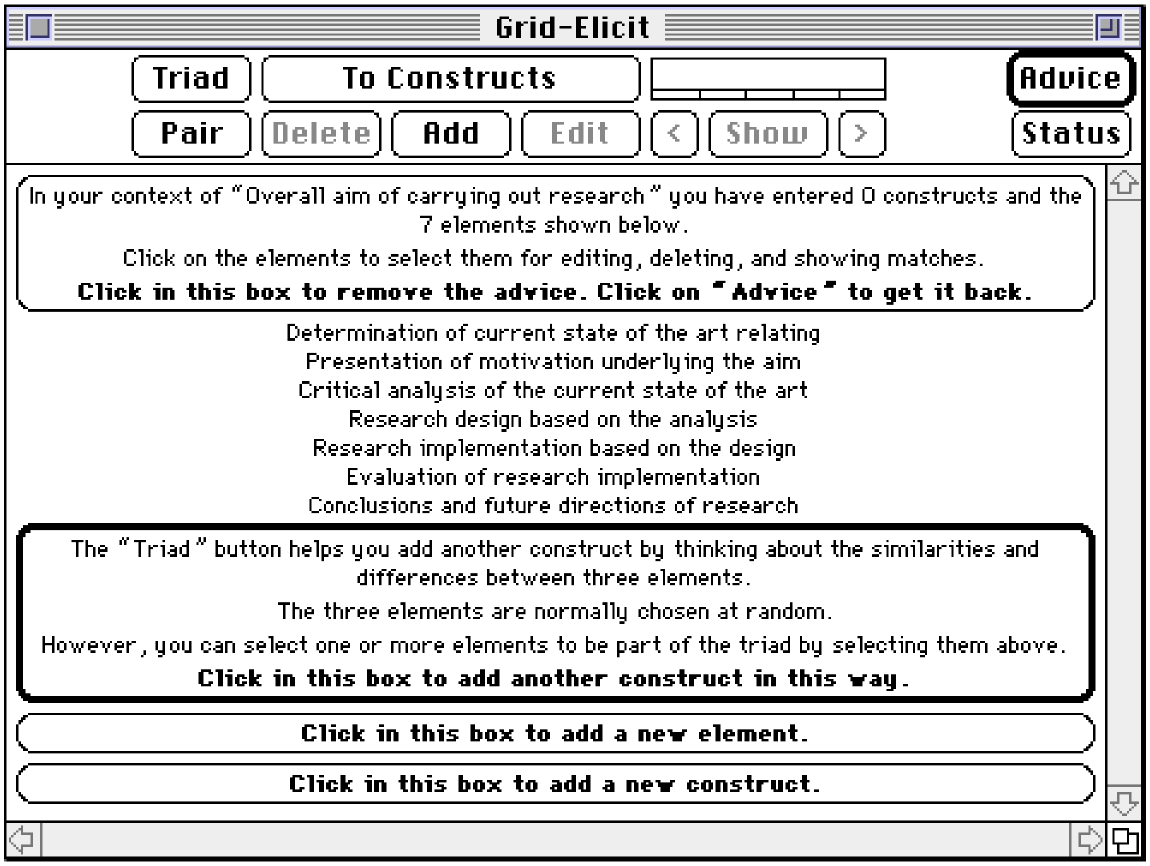
Elements
Determination of current state of the art relating to the aim
Presentation of motivation underlying the aim
Critical analysis of the current state of the art
Research design based on the analysis

Annotation

Figure 15 Automatic creation of initial elements for a repertory grid

The node type “Grid” has been predefined to have a special annotation card with associated scripts. The user has entered a “Grid” node at the middle left of the concept map and linked it to the relation node “requires” which itself is a link from the concept “Overall aim of carrying out research” to a list of requirements. It is this list that makes an interesting set of elements to be distinguished along various dimensions using repertory grid elicitation. When the user selects the “Annotation” option for the “Grid” node, the HyperCard stack is programmed to generate the specialist annotation card shown at the bottom of Figure 5.

When the user clicks on the “Get Data” button at the bottom left the HyperCard script requests the nodes linked to the “Grid” node from KMap and uses these to determine the context for a grid together with an initial set of elements. When the user clicks on the “Start Grid” button the HyperCard script creates a repertory grid file in a standard data format and opens it in the RepGrid (CPCS, 1993) repertory grid elicitation program as shown in Figure 16. The RepGrid program itself supports annotation links to HyperCard and to expert system shells (Gaines, Rappaport and Shaw, 1992; Shaw, Gaines and Linster, 1994), thus allowing a highly integrated knowledge acquisition tool to be created from separate components each with well-defined specialist functionality.



In your context of “Overall aim of carrying out research” you have entered 0 constructs and the 7 elements shown below.

Click on the elements to select them for editing, deleting, and showing matches.

Click in this box to remove the advice. Click on “Advice” to get it back.

Determination of current state of the art relating
Presentation of motivation underlying the aim
Critical analysis of the current state of the art
Research design based on the analysis
Research implementation based on the design
Evaluation of research implementation
Conclusions and future directions of research

The “Triad” button helps you add another construct by thinking about the similarities and differences between three elements.

The three elements are normally chosen at random.

However, you can select one or more elements to be part of the triad by selecting them above.

Click in this box to add another construct in this way.

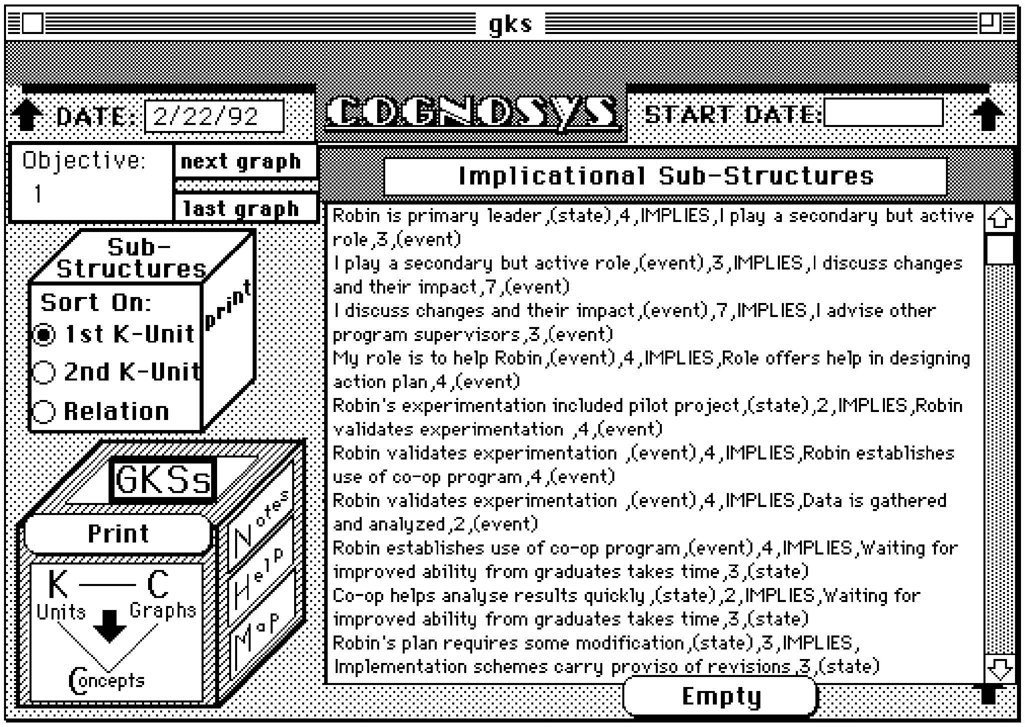
Click in this box to add a new element.

Click in this box to add a new construct.

Figure 16 Repertory grid elicitation with elements from concept map

10 Automatic Generation of Concept Maps

The concept maps shown in most of the examples so were all generated manually by a user entering data into the KMap graphic editor. The example of Figure 9 shows how inference may be used with formal semantic nets to generate additional nodes automatically. It is also possible to have concept maps that are in whole or major part automatically generated through some computational process. For example, Woodward's (1990) Cognosys tool for knowledge acquisition from text supports the analysis of protocols in terms of Graesser and Clark's (1985) linguistically derived "general knowledge structures" (GKS). Figure 17 shows text being split up in Cognosys and annotated with the GKS typology and relations.



DATE: 2/22/92

START DATE:

Objective: 1

Sort On: [✓ 1st K-Unit] 2nd K-Unit Relation

Implicational Sub-Structures

Robin is primary leader,(state),4,IMPLIES,I play a secondary but active role,3,(event)
I play a secondary but active role,(event),3,IMPLIES,I discuss changes and their impact,7,(event)
I discuss changes and their impact,(event),7,IMPLIES,I advise other program supervisors,3,(event)
My role is to help Robin,(event),4,IMPLIES,Role offers help in designing action plan,4,(event)
Robin's experimentation included pilot project,(state),2,IMPLIES,Robin validates experimentation ,4,(event)
Robin validates experimentation ,(event),4,IMPLIES,Robin establishes use of co-op program,4,(event)
Robin validates experimentation ,(event),4,IMPLIES,Data is gathered and analyzed,2,(event)
Robin establishes use of co-op program,(event),4,IMPLIES,Waiting for improved ability from graduates takes time,3,(state)
Co-op helps analyse results quickly ,(state),2,IMPLIES,Waiting for improved ability from graduates takes time,3,(state)
Robin's plan requires some modification,(state),3,IMPLIES, Implementation schemes carry proviso of revisions,3,(state)

Figure 17 Development of knowledge structures from interview text in Cognosys

Cognosys then exports this annotated text to KMap which automatically draws it as shown in Figure 18 as a set of nodes whose types correspond to the GKS types and relations, and whose content is the fragments of the text being analyzed. The algorithm for the graph layout was derived from one described by Watanabe (1989) which essentially uses a set of rules designed to minimize line crossings.

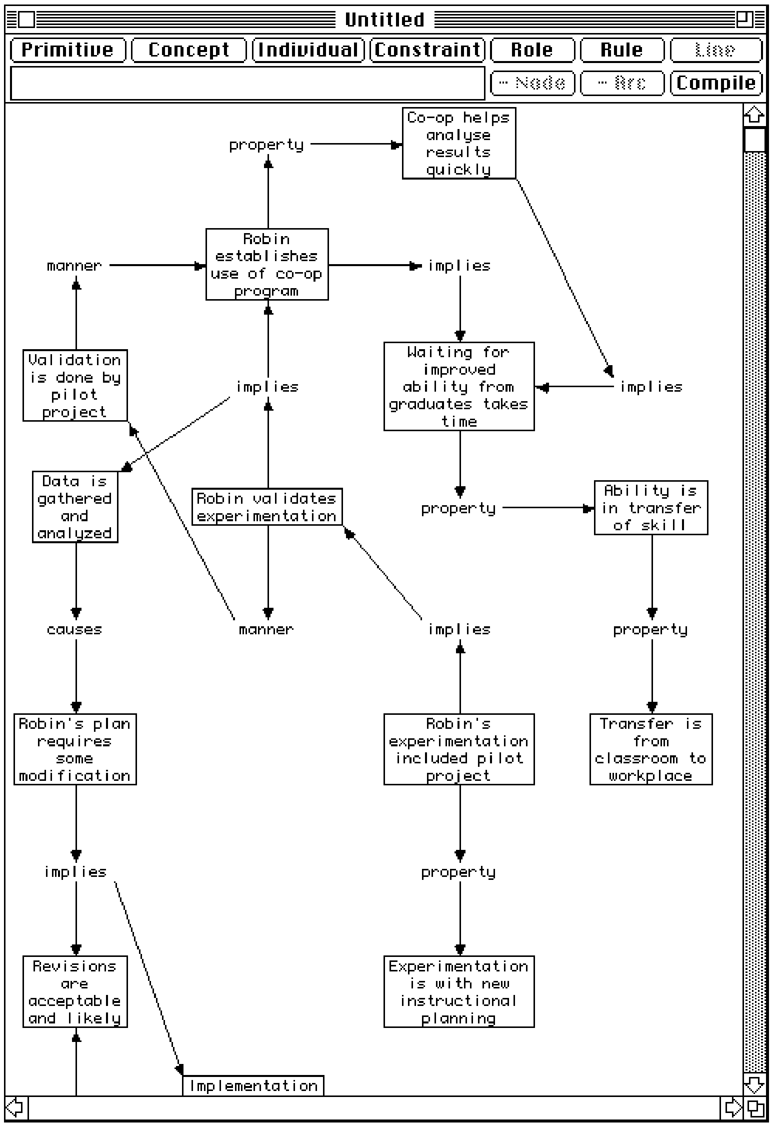

Untitled
Primitive
Concept
Individual
Constraint
Role
Rule
Line
Node
Arc
Compile
Co-op helps analyse results quickly
property
Robin establishes use of co-op program
manner
implies
Validation is done by pilot project
implies
Waiting for improved ability from graduates takes time
implies
Data is gathered and analyzed
Robin validates experimentation
property
Ability is in transfer of skill
causes
manner
implies
property
Robin's plan requires some modification
Robin's experimentation included pilot project
Transfer is from classroom to workplace
implies
property
Revisions are acceptable and likely
Experimentation is with new instructional planning
Implementation

Figure 18 Automatic layout of knowledge structures from Cognosys in KMap

It is also possible to develop concept maps by fully automatic text analysis. Figure 19 shows a map generated automatically through analysis of the co-occurrence of words in sentences (Gaines and Shaw, 1994), a technique commonly used in information retrieval systems (Callon, Law and Rip, 1986). The document analyzed is one on *The Technical Concept of IMS* (Tomiyama, 1992) that played a major role in the design of the international Intelligent Manufacturing Systems (IMS) research program. The document is treated as a set of entities which are sentences whose features are the words they contain. Rules are derived using empirical induction in which the premise is that if one word occurs in a sentence then the conclusion is that another will occur. The graph shows the links from premises to conclusions derived in this way.

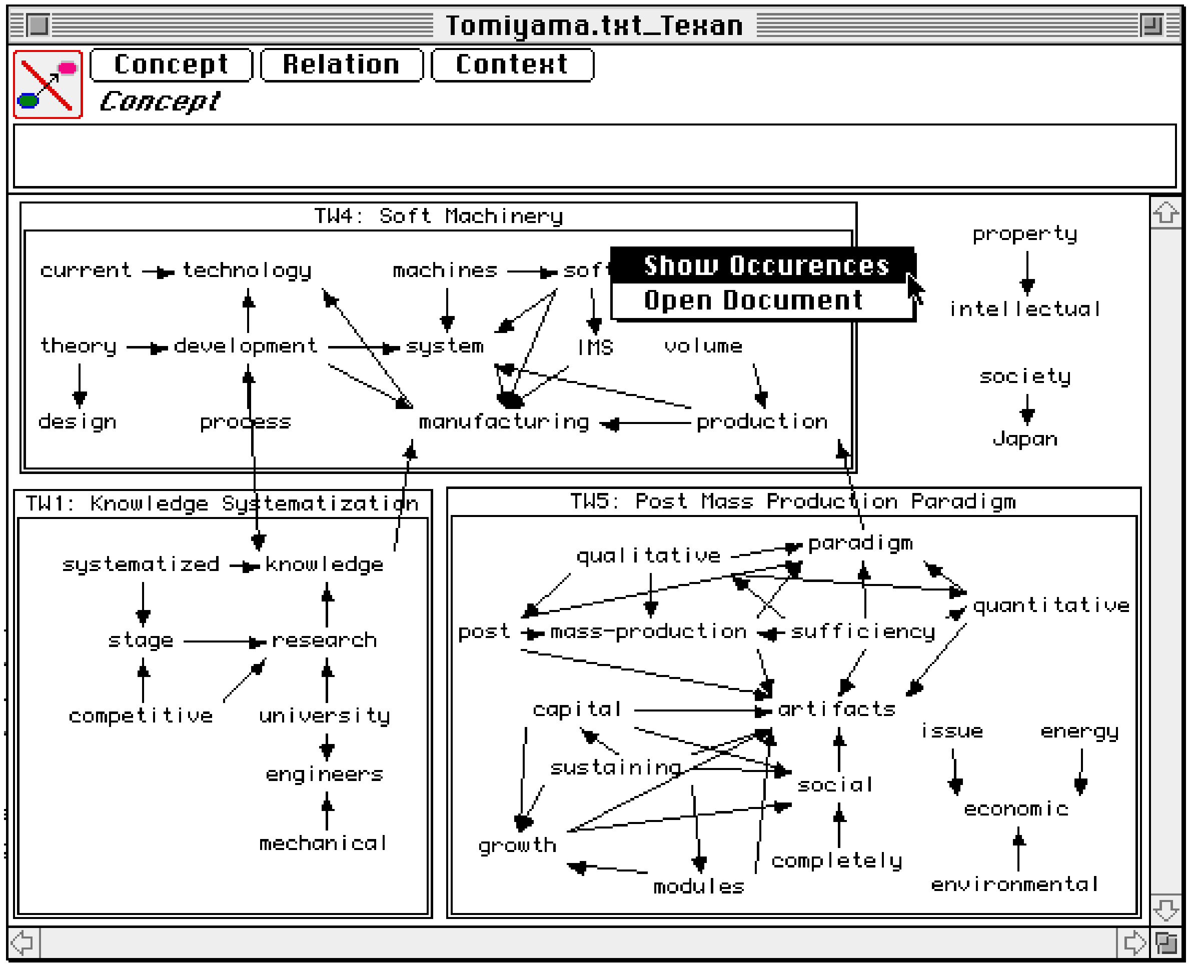


Figure 19 Concept map of GNOSIS project derived by text analysis

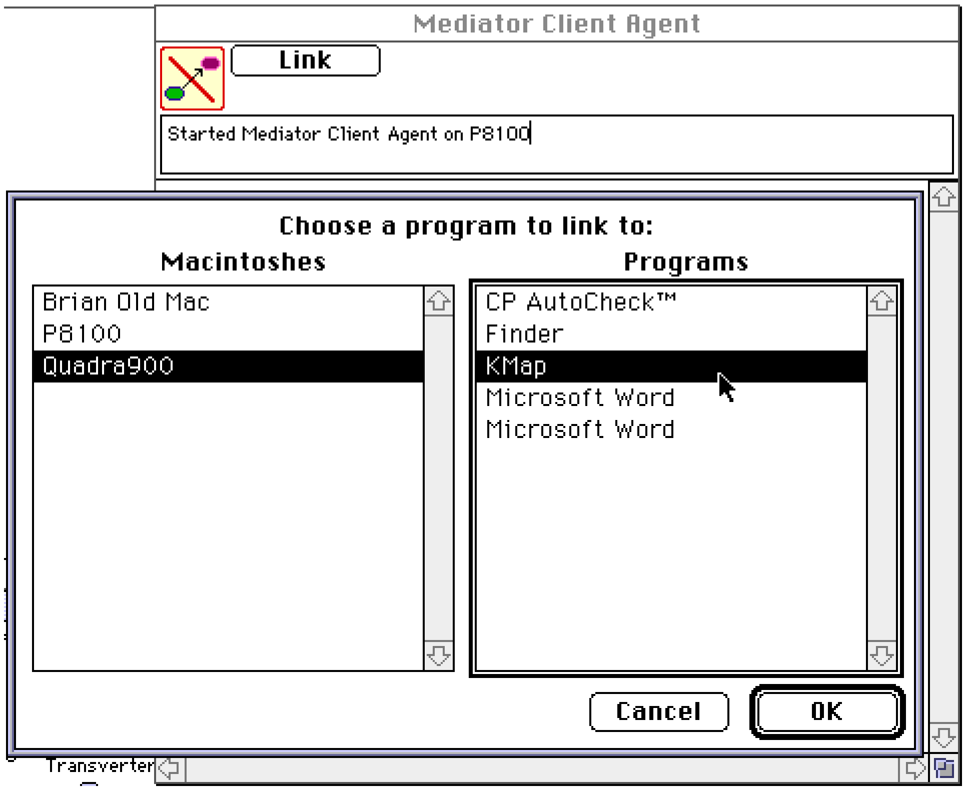
The initial output was a digraph consisting of one major connected component and some minor ones which correspond to significant topics such as intellectual property rights that did not directly relate to the socio-technical issues. The user noted that the major component itself consisted of 3 loosely connected sub-components, and added the context boxes shown to distinguish and name these parts. The significance of these parts is that they correspond to 3 of the 5 technical work packages (TW's) of the IMS GNOSIS research program. What is particularly significant is the missing work packages, TW2 concerned with product configuration management systems, and TW3 concerned with configurable production systems. These were added to the research program during its formation through amalgamation of interests with other potential proponents of IMS test cases. These packages link technically to the knowledge systematization activities on the left of Figure 19 but are neutral to the major issues of the IMS program on the right. Thus, the concept map produced by automatic text analysis provides insights into the underlying conceptual structure of the research program defined in the document analyzed.

A popup menu associated with each word provides hypertext links to a list of occurrences of that word in context, and to the original document. Additional links may also be added to other documents such as the progress reports on the associated projects, so that the concept map may be used as a project management tool.

11 Collaborative Access to Concept Maps

The general semi-formal diagrams of concept maps provide a very attractive basis for collaborative brain-storming and discussion. We have used the initial, non-networked version of KMap in learning situations with small groups developing conceptual frameworks collaboratively by working together on the same concept map. We have also found it useful for one learner to take the concept map of another, modify and extend it, and then enter into a group discussion about what changes they have made, and why. We have also linked KMap to repertory grid tools so that nodes representing a set of related examples in a concept map may be used to elicit a relevant conceptual framework through grid elicitation (Shaw and Gaines, 1992).

The scripting and communication capabilities of the class library in which KMap is implemented make it possible to provide explicit support for both synchronous and asynchronous collaboration over local and wide area networks. For example, Figure 20 shows the Mediator (Gaines and Norrie, 1994) collaborative concept mapping system when a user has opened a blank concept map with a script supporting client agent functionality. The user clicks on the "Link" button and a dialog box appears allowing her to select KMap running on her colleague's machine. When she does so her KMap requests a server agent running on the other machine to transmit a concept map of its top level data structures as shown in Figure 21.

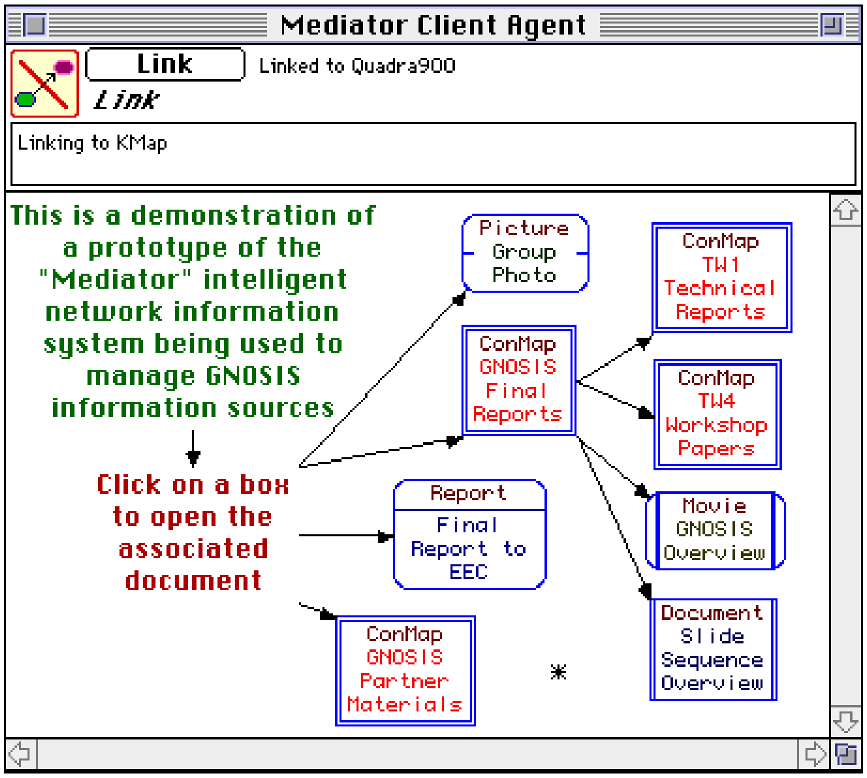


Started Mediator Client Agent on P8100

Macintoshes	Programs
Brian Old Mac	CP AutoCheck™
P8100	Finder
Quadra900	KMap
	Microsoft Word
	Microsoft Word

Figure 20 Linking concept map clients and servers across a network

The cursor has changed from an arrow to a star at the lower right of Figure 21 to indicate that the concept map is not editable. The user can double-click in the window to make the map editable, changing the cursor back to an arrow. If the same concept map is open on the server machine this action will automatically make the remote map non-editable. Changes made to the editable map are transmitted to any linked ones.


Mediator Client Agent

Linked to Quadra900

Linking to KMap

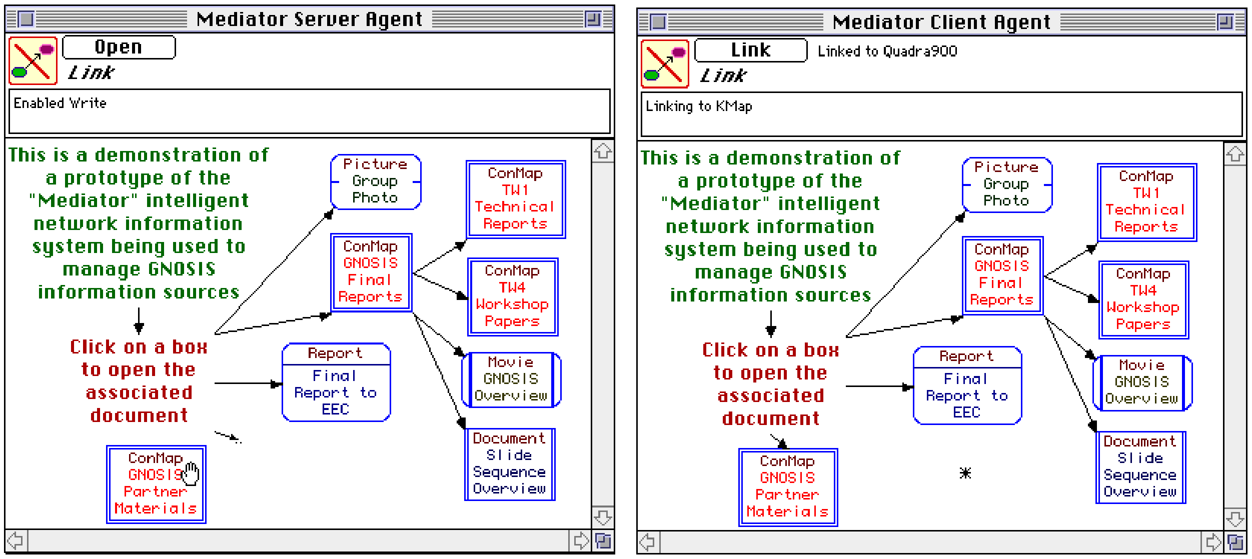
This is a demonstration of a prototype of the "Mediator" intelligent network information system being used to manage GNOSIS information sources

Click on a box to open the associated document

Link
Link
Picture Group Photo
ConMap TW1 Technical Reports
ConMap GNOSIS Final Reports
ConMap TW4 Workshop Papers
Report
Final Report to EEC
Movie GNOSIS Overview
Document Slide Sequence Overview
ConMap GNOSIS Partner Materials

Figure 21 Concept map fetched from a server agent

Figure 22 left shows a node of the map on the server being dragged to the left, and the change occurring in the linked map on the right.


Mediator Server Agent

Enabled Write

This is a demonstration of a prototype of the "Mediator" intelligent network information system being used to manage GNOSIS information sources

Click on a box to open the associated document

Mediator Client Agent

Linked to Quadra900

Linking to KMap

This is a demonstration of a prototype of the "Mediator" intelligent network information system being used to manage GNOSIS information sources

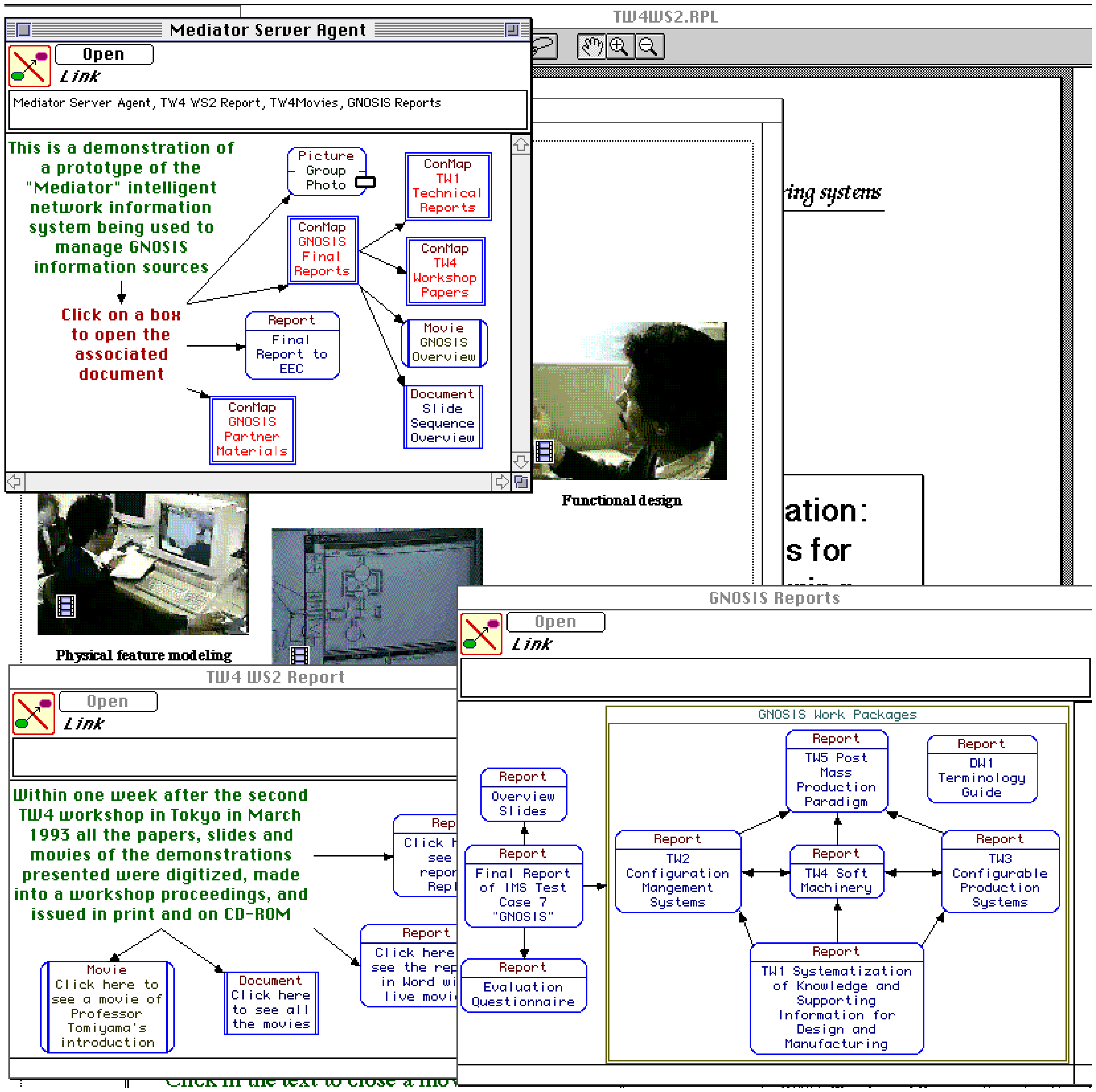
Click on a box to open the associated document

Open
Link
Link
Link
Picture Group Photo
ConMap TW1 Technical Reports
ConMap GNOSIS Final Reports
ConMap TW4 Workshop Papers
Report
Final Report to EEC
Movie GNOSIS Overview
Document Slide Sequence Overview
ConMap GNOSIS Partner Materials

Figure 22 Changes to the concept map on one machine are transmitted to the linked map

Figure 23 shows a screen dump of a sequence of material being accessed from the concept maps in Figure 22. The map is shown at the upper left currently write-disabled, and the cursor has changed to a button as the user mouses over the “Group Photo” node. Clicking at this point will display the photograph in a separate window. The user has already clicked on the node “GNOSIS

Final Reports” to open the concept map shown at the lower right. This has a node for each report, and clicking on one will open the appropriate report. The user has already clicked on the “TW4 Workshop Papers” node in the concept map at the top left, and opened the relevant concept map at the bottom left. She has then clicked on the node “Click here to see the report in Replica”, and opened the report visible at the back on the right. She has also clicked on the “Click here to see all the movies” node and opened the document visible behind the concept maps which displays QuickTime movies, any of which can be played by double clicking upon it.


TW4WS2.RPL
Mediator Server Agent
Open
Link
Mediator Server Agent, TW4 WS2 Report, TW4Movies, GNOSIS Reports
Picture Group Photo
ConMap TW1 Technical Reports
ConMap GNOSIS Final Reports
ConMap TW4 Workshop Papers
Click on a box to open the associated document
Report Final Report to EEC
Movie GNOSIS Overview
ConMap GNOSIS Partner Materials
Document Slide Sequence Overview
Functional design
Physical feature modeling
TW4 WS2 Report
Open
Link
Rep
Click h see repor Repl
Report Click here see the rep in Word wi live movi
Movie Click here to see a movie of Professor Tomiyama's introduction
Document Click here to see all the movies
GNOSIS Reports
Open
Link
GNOSIS Work Packages
Report Overview Slides
Report Final Report of IMS Test Case 7 "GNOSIS"
Report Evaluation Questionnaire
Report TW5 Post Mass Production Paradigm
Report DW1 Terminology Guide
Report TW2 Configuration Mangement Systems
Report TW4 Soft Machinery
Report TW3 Configurable Production Systems
Report TW1 Systematization of Knowledge and Supporting Information for Design and Manufacturing

This is a demonstration of a prototype of the "Mediator" intelligent network information system being used to manage GNOSIS information sources

Within one week after the second TW4 workshop in Tokyo in March 1993 all the papers, slides and movies of the demonstrations presented were digitized, made into a workshop proceedings, and issued in print and on CD-ROM

Figure 23 Accessing networked hypermedia materials through layered concept maps

Thus, the scripting and communication capabilities of KMap may be used to create a collaborative environment for the editing of concept maps and their application to tasks such as indexing distributed hypermedia materials. The overall architecture of the Mediator system supporting the application illustrated in Figure 23 is shown in Figure 24. The “Server Agent” and

"Client Agent" are implemented in KMap scripts. A server agent manages the relevant files at its site, providing a top-level concept map indexing the files through layers of concept maps that access other concept maps or files specific to some other application. A client agent links to a server agent at a remote site, and emulates the behavior of the remote server agent at the local site.

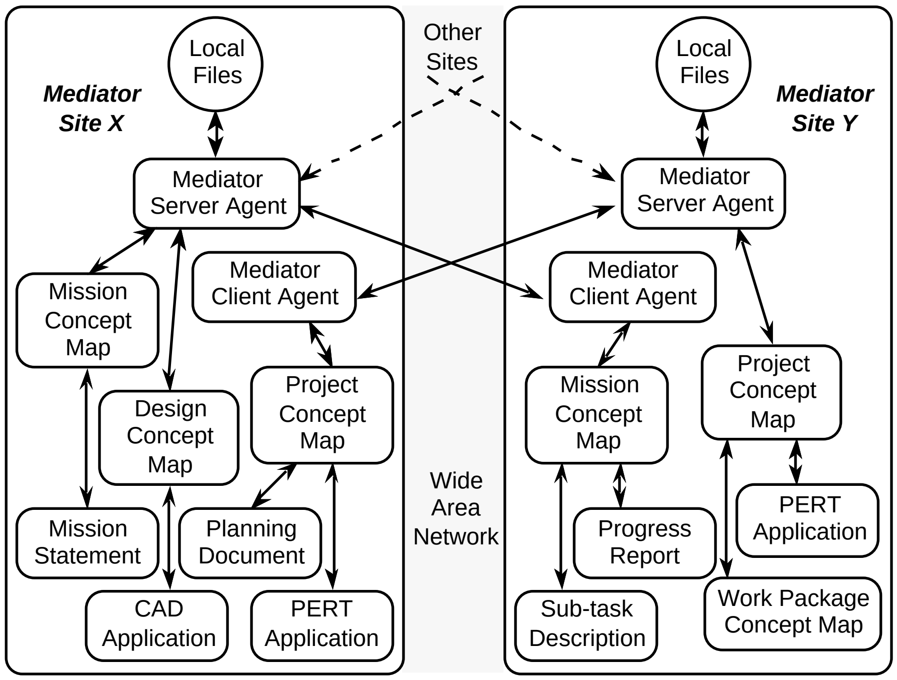


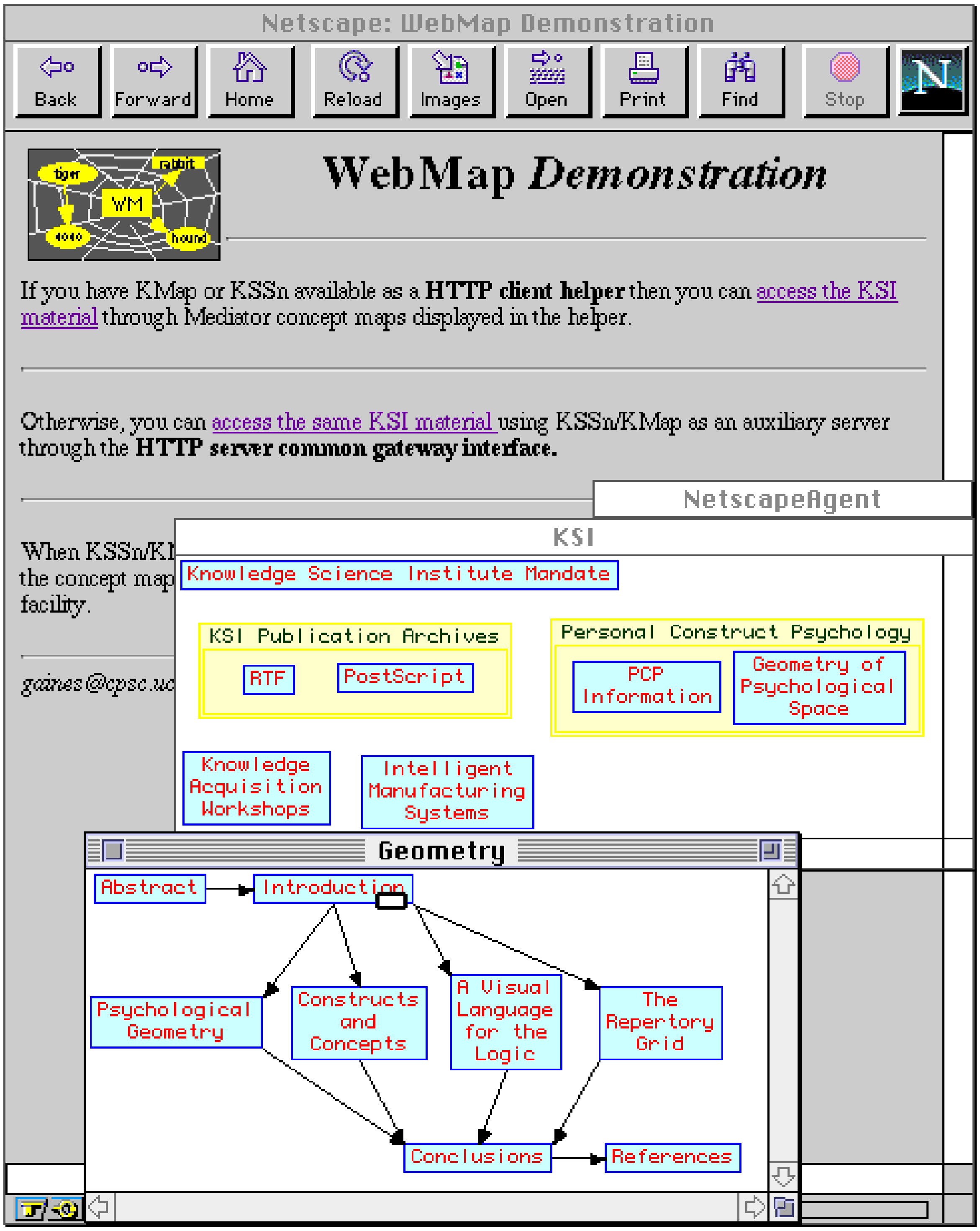
Figure 24 Mediator architecture supporting networked hypermedia access

When application specific files are fetched they are opened in the relevant application. If the application has functionality to support integration with other applications then it may pass messages back to the client and server agents. An example is the integration with Netscape (1995) described in the following section.

12 Concept Map Helpers on World-Wide Web

One of the major objectives of our research in the last two years has been to port all our interactive systems, currently on personal computers and local area networks, to operate effectively in the open, wide area networking of the Internet, and, wherever possible, to integrate seamlessly with the World Wide Web (WWW, Berners-Lee, Cailliau, Luotonen, Nielsen and Secret, 1994). However, the graphic primitives necessary for interactive concept maps are not provided as part of the HTML widget set, and hence it is a challenge to support concept maps on the web.

The scripting capabilities make it possible to operate KMap as a WWW *client helper*, capable of accepting concept maps brought across the web by a browser such as Netscape, and of requesting arbitrary files, including concept maps, to be fetched through messages sent from KMap scripts to the WWW browser. Figure 25 shows an HTML document in Netscape which has links to KMap documents. When the user clicks on the hypertext link "access the KSI material" at the top of the screen, Netscape fetches a KMap file "KSI" which it passes to KMap which opens and displays it as shown in the KMap window at the middle right.


Netscape: WebMap Demonstration

WebMap *Demonstration*

If you have KMap or KSSn available as a **HTTP client helper** then you can access the KSI material through Mediator concept maps displayed in the helper.

Otherwise, you can access the same KSI material using KSSn/KMap as an auxiliary server through the **HTTP server common gateway interface.**

When KSSn/K
the concept map
facility.

gaines@cpsc.uc

NetscapeAgent

KSI

Knowledge Science Institute Mandate

KSI Publication Archives: RTF, PostScript

Personal Construct Psychology: PCP Information, Geometry of Psychological Space

Knowledge Acquisition Workshops

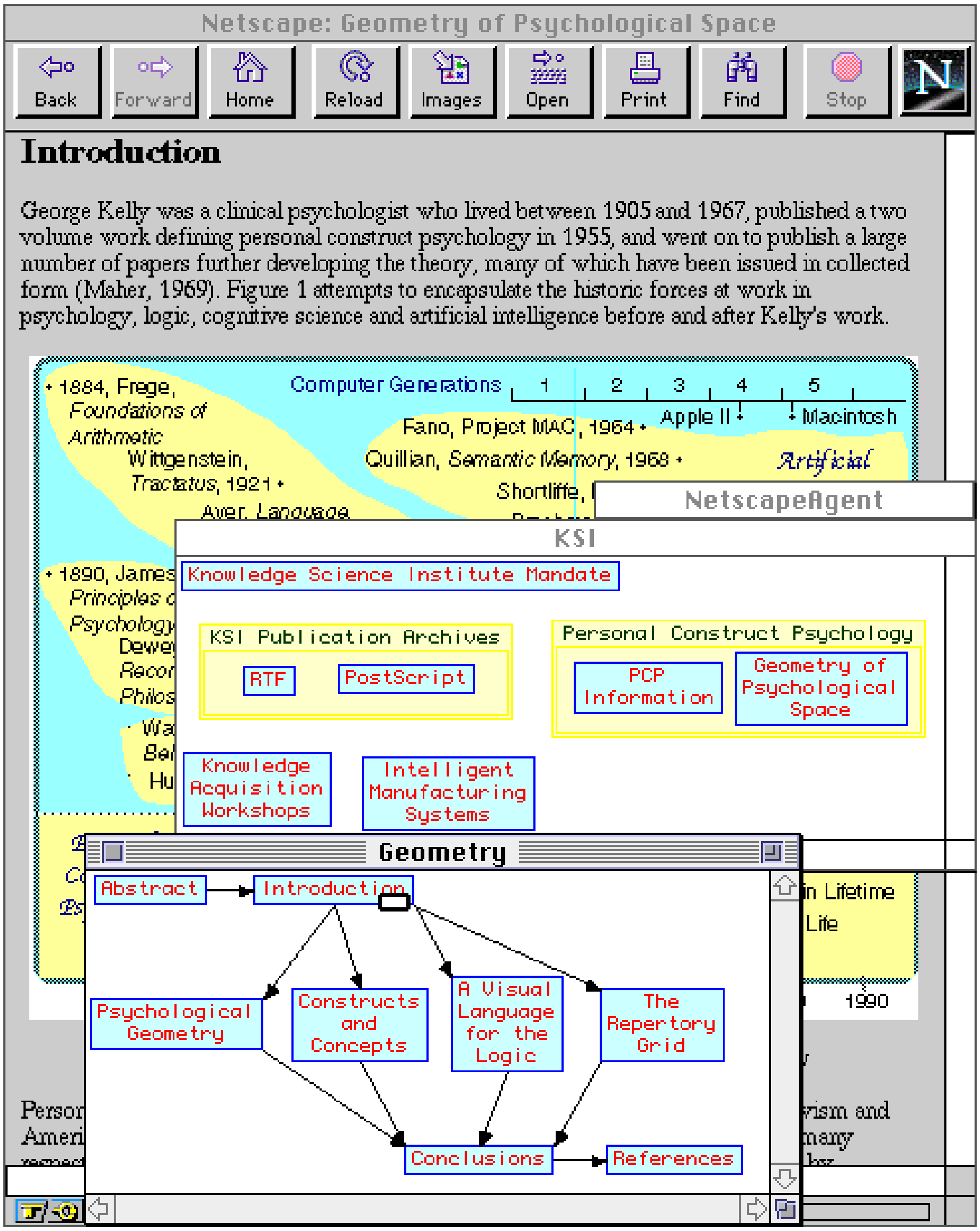
Intelligent Manufacturing Systems

Geometry

Abstract → Introduction → Psychological Geometry, Constructs and Concepts, A Visual Language for the Logic, The Repertory Grid → Conclusions → References

Figure 25 KMap acts a client helper application to the Netscape browser

Clicking on the node “Geometry of Psychological Space” at the right of this concept map causes KMap to send a message to Netscape requesting that the file with uniform resource locator (URL) “http://ksi.cpsc.ucalgary.ca/WebMap/Geometry.KSS” be fetched. This is a concept map of the structure of an article provided by its author that opens in another KMap window shown at bottom right. Clicking on the node “Introduction” at the top of this concept map causes KMap to send a message to Netscape requesting that the file with URL “http://ksi.cpsc.ucalgary.ca/PCP/PCPIntro.html#3” be fetched. This is a section of an HTML document that Netscape then displays in its own window as shown in Figure 26. Clicking on other nodes in the concept map causes Netscape to navigate to different sections of this document.


Netscape: Geometry of Psychological Space

Introduction

George Kelly was a clinical psychologist who lived between 1905 and 1967, published a two volume work defining personal construct psychology in 1955, and went on to publish a large number of papers further developing the theory, many of which have been issued in collected form (Maher, 1969). Figure 1 attempts to encapsulate the historic forces at work in psychology, logic, cognitive science and artificial intelligence before and after Kelly's work.

Figure 26 KMap fetching a file in Netscape

Note the “Netscape Agent” concept map window appearing just behind the “KSI” concept map in Figures 25 and 26. It is this map that has the scripts that control the behavior of the other concept maps. The maps fetched from remote sites are passive data structures that in themselves can only display on the local machine. They are given active functionality through KMap passing messages from them to the “Netscape Agent” which has registered itself with KMap as an active agent. Thus, no programs that could have adverse effects on the operation of the local machine are loaded from remote sites. Clearly, at some stage the agent itself has to be acquired, but this need only be done once, and it can be assessed for adverse behaviors by examining its script.

This application of KMap as a helper to a World-Wide web browser allows much of the multimedia file sharing of Mediator illustrated in the previous section to be achieved by the integration of different applications, with KMap no longer having to manage TCP/IP communications. It is desirable in system development to achieve the functionality required by using as many standard and widely available components as possible, and minimizing any additional code that has to be written. Eventually it should become possible to achieve the functionality of KMap using web scripting languages such as Java (Sun, 1995) which are designed to download extensions to browsers while enforcing strong protection against adverse behavior.

13 Concept Map Creation through World-Wide Web Browsing

Netscape allows an external application to register itself to receive a range of information about user interaction with Netscape. This enables a KMap agent to receive a trace of user interactions with WWW, build a concept map of linked documents, and use this to provide access to those documents—a graphic “hot-list.” Figure 27 shows such a hot list being developed and used. The “Netscape Agent” in KMap has sent a “Register URL Message” to Netscape requesting that it be informed when Netscape shows a document. Each time that Netscape does so it send KMap a message giving the URL of the document, the URL of the referring document if there was one and the Window in which the document is being shown.

The “Netscape Agent” script stores the URLs and displays the name of the window in the text box near the top of the concept map window. If the user clicks on the popup menu button, the script creates a new node of the type specified in the popup menu. It stores the name of the window in the visible content of the node because this name is the <TITLE> field of the HTML document. It stores the URL of the document in the invisible content of the node so that clicking on the node may be used to request Netscape to fetch the document. If there is already a node for the referring document then the script links it with an arrow to the new node. The overall effect is to create a hot list of locations visited that can be used to access these locations again.

All of the agent functionality is available in new concept maps that are created. In Figure 27 the user has created a new concept map called “Web66” to develop a map of materials accessible from the Web66 site. All the nodes in the map shown were created by browsing in Netscape starting from the Web66 site.

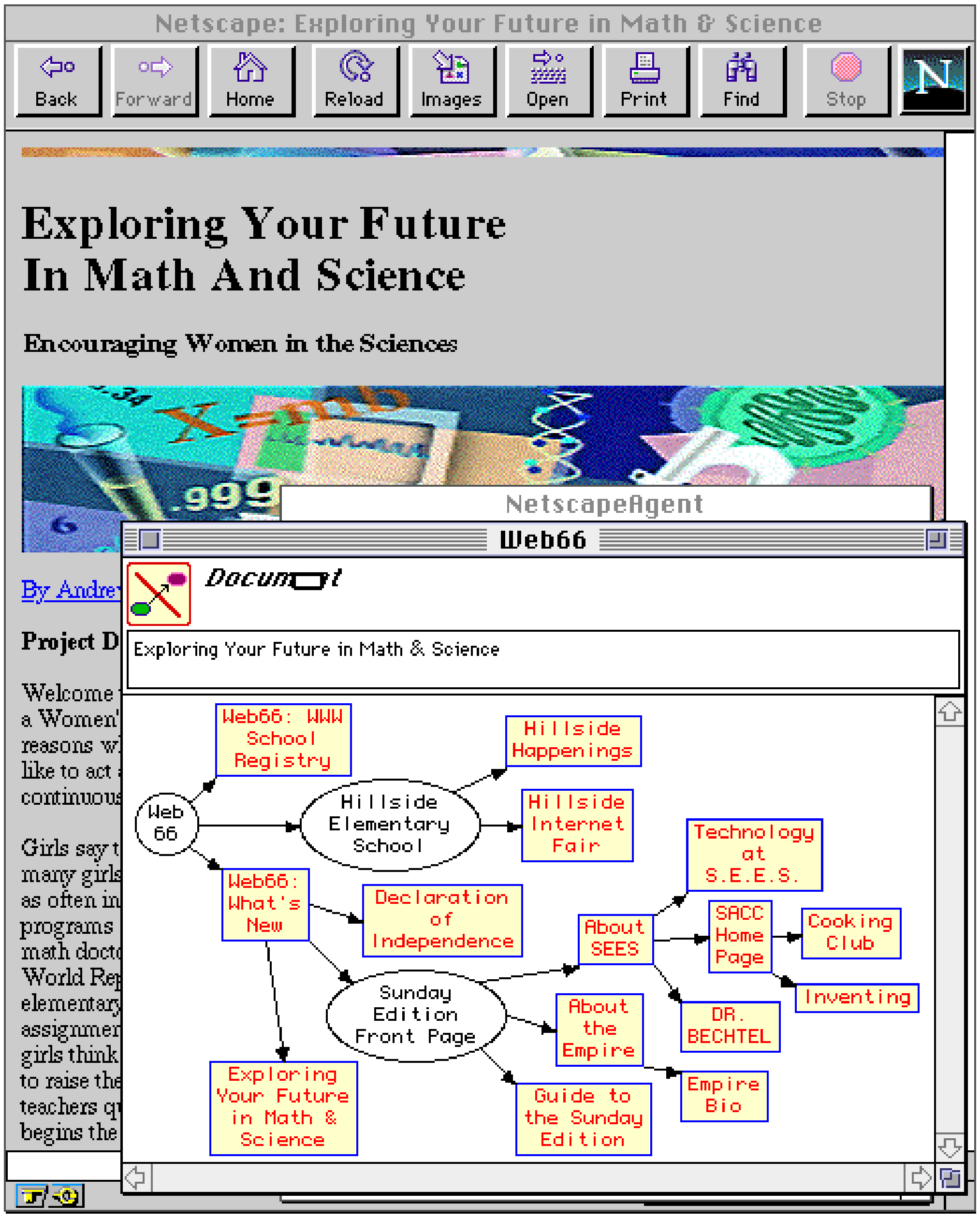

Netscape: Exploring Your Future in Math & Science
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Exploring Your Future
In Math And Science
Encouraging Women in the Sciences
NetscapeAgent
Web66
Document
Exploring Your Future in Math & Science
Web66: WWW School Registry
Hillside Happenings
Web 66
Hillside Elementary School
Hillside Internet Fair
Technology at S.E.E.S.
Web66: What's New
Declaration of Independence
About SEES
SACC Home Page
Cooking Club
Inventing
Sunday Edition Front Page
About the Empire
DR. BECHTEL
Exploring Your Future in Math & Science
Guide to the Sunday Edition
Empire Bio

Figure 27 KMap developing a graphic "hot list" from Netscape browsing

Nielsen (1995) reports a related development of an "overview diagram" in the KJ-Editor developed by Ohiwa and Kawai. The overview diagram is constructed by the user who for each page visited in the World-Wide Web browser can decide whether to add it to the map window. It is significant to note how simple it is to add this type of functionality to an open architecture browser such as Netscape using a scriptable concept mapping tool rather than developing a special-purpose application. The Netscape Agent script took some 2 hours to write and check in operation.

Nielsen (1995) also emphasizes the limits of World-Wide Web browsers in supporting advanced graphical user interaction techniques because, even though it is possible to have multiple windows, the windows do not communicate and thus cannot be used to support advanced hypertext features that would require the coordination of the content of several windows. It is symptomatic of the speed with which World-Wide Web technology is developing that it is now simple with Netscape and KMap to provide such coordination. The use of script languages in helper applications coordinating open architecture browsers provides the capability to experiment with elaborate hypermedia models.

14 Embedded Clickable Concept Map Images on World-Wide Web

KMap is currently implemented only for the Apple Macintosh and hence can act as a client helper only on Macintosh computers. Ports to Windows and Motif are being developed which will make KMap helpers available on all major platforms. However, there will always be users who do not have, or do not want to use, the helpers but where it is appropriate to provide non-editable concept maps as *clickable maps* in HTML documents. Hence KMap has also been interfaced as an auxiliary HTTP server to allow the same concept maps to be used as clickable maps.

The auxiliary server also allows the "Netscape Agent" at the browser to upload the concept maps created locally to the server. As shown in Figure 28, clicking in an empty part of a concept map and holding the mouse down for a period brings up a popup menu with a "Send" option. This menu is created by the agent when it receives the mouse down message from the map. When the agent receives the "Send" selection it uploads the concept map data to the server using Netscape's capability to accept an "Open URL" event with POST parameters. The map data is automatically stored under the same concept map name in an upload directory named "X" using an auxiliary HTTP server written in the same class library as KMap operating through a common gateway interface to a WebSTAR (StarNine, 1995) HTTP server. The enforced "X" root directory prevents this capability being used to overwrite files in areas other than that reserved for uploading. More specific protection such as password control can be enforced if required.

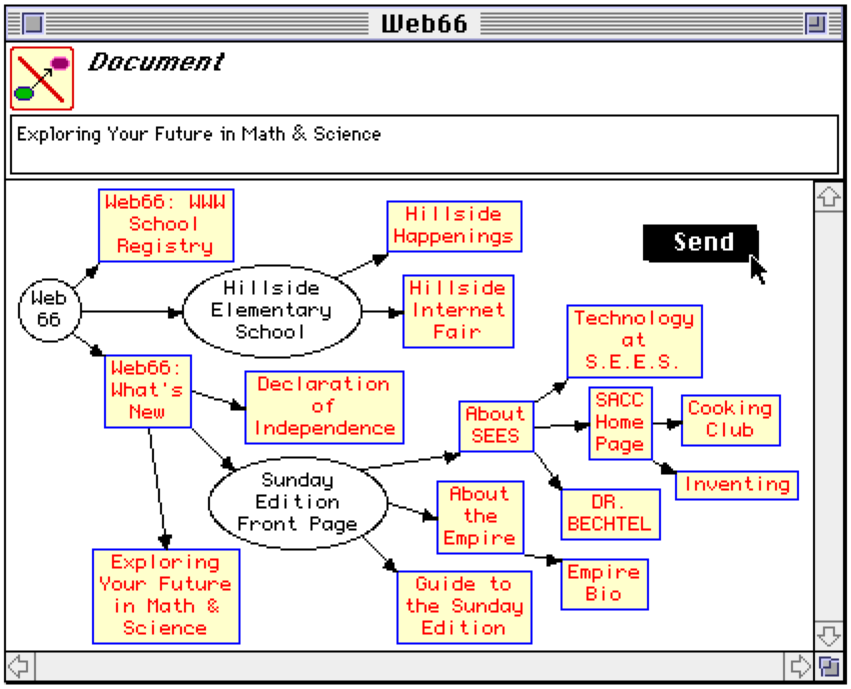

Web66
Document
Exploring Your Future in Math & Science
Web66: WWW School Registry
Hillside Happenings
Send
Web 66
Hillside Elementary School
Hillside Internet Fair
Technology at S.E.E.S.
Web66: What's New
Declaration of Independence
About SEES
SACC Home Page
Cooking Club
Inventing
Sunday Edition Front Page
About the Empire
DR. BECHTEL
Exploring Your Future in Math & Science
Guide to the Sunday Edition
Empire Bio

Figure 28 KMap uploading the concept map to the HTTP server

KMap operating as auxiliary server provides two functions to utilize the concept maps which have been uploaded. First, it will download concept maps as clickable maps in GIF format. Second, it will receive clicks on these maps and cause the same file to be sent as if the concept map had been clicked locally in a KMap helper. Figure 29 shows the uploaded concept map of Figure 27 being fetched using the URL “http://tiger.cpsc.ucalgary.ca/WMClick/:X:K66.k” which is recognized by WebStar as a common gateway interface (CGI) *action* to be passed to KMap.

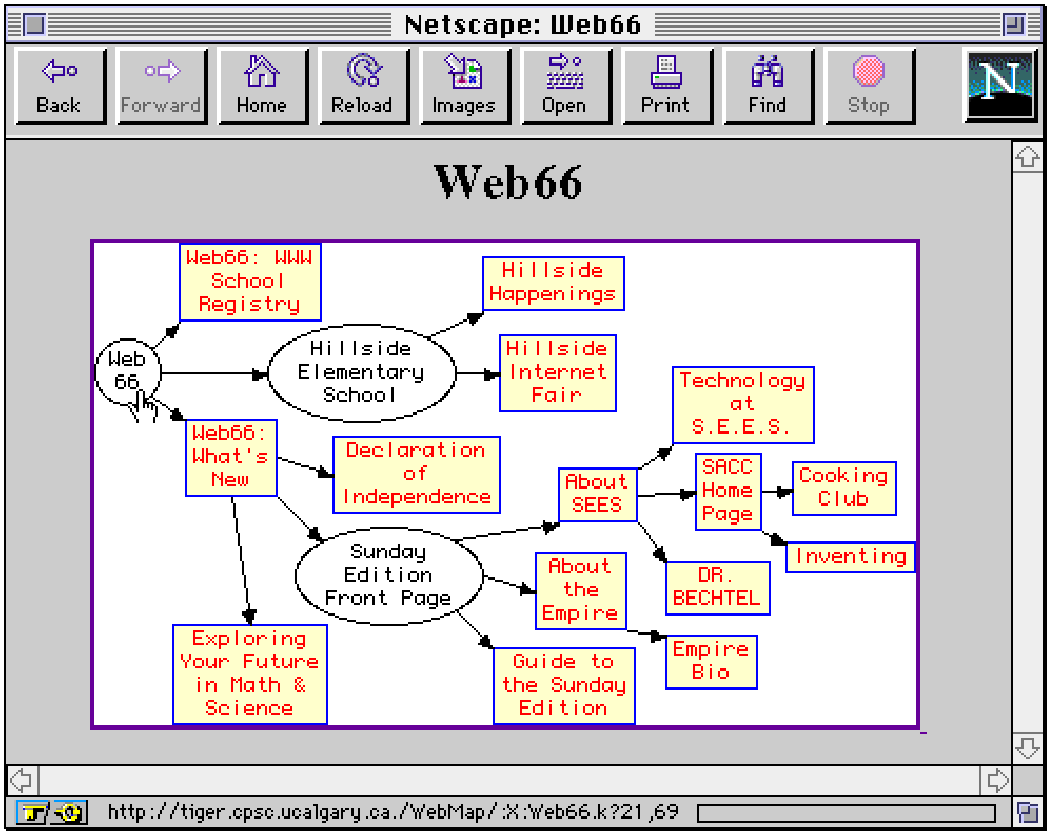


Figure 29 KMap acting as an HTTP server for clickable maps

KMap returns the HTML shown in Figure 30 with the name of the concept map as a title and an embedded call for an image of the concept map. The image URL is again interpreted by WebStar as an action passed to KMap which, since there are no search arguments, loads the concept map, converts its image to a GIF and returns it as an HTTP message. Clicking on the node “Web66” sends the coordinates of the point clicked back to KMap on the server as search arguments, and KMap then takes the same action as if the map had been clicked locally in a helper, in this case to redirect Netscape to the URL for the Web66 home page.

```
<HTML><HEAD><TITLE>K66</TITLE></HEAD>
<BODY><H1 ALIGN="CENTER">K66</H1>
<P ALIGN="CENTER">
<A HREF="http://tiger.cpsc.ucalgary.ca./WebMap/:X:K66.k">
<IMG SRC="http://tiger.cpsc.ucalgary.ca./WebMap/:X:K66.k" ISMAP>
</A></BODY></HTML>
```

Figure 30 HTML for KMap acting as an HTTP server for clickable maps

Thus KMap supports the use of concept maps on the World-Wide Web through client helpers and through servers in an integrated way. Concept maps created in KMap for use in conjunction with Netscape as a local helper can be automatically uploaded and used in the same was as clickable maps through KMap as an auxiliary server. In particular, this allows concept maps to be used by browsers running on platforms other than the Macintosh.

It can be seen from the HTML of Figure 30 that any number of concept maps may be embedded as clickable maps in an HTML document by incorporating the “<A HREF...” and “<IMG SRC..” pairs for each map. This enables much of the functionality of the active document system illustrated in Figure 6 to be recreated on World-Wide Web. For example, knowledge bases may be embedded in documents as semantic networks and used as part of expert systems. The only capability missing is for the user to be able to edit the knowledge base within the document. In the near future it should be possible to implement this also by rewriting the concept mapping tools in a web code tool such as Java.

15 Conclusions

Concept maps have been used in education, policy studies and the philosophy of science to provide a visual representation of knowledge structures and argument forms. They provide a complementary alternative to natural language as a means of communicating knowledge. In many disciplines various forms of concept map are already used as formal knowledge representation systems, for example: semantic networks in artificial intelligence, bond graphs in mechanical and electrical engineering, CPM and PERT charts in operations research, Petri nets in communications, and category graphs in mathematics. This article has described the design and applications of an open architecture concept mapping tool, shown how it may be used to support a wide range of applications and disciplines, with emphasis on hypermedia collaborative activities through networks and the World-Wide Web.

User experience with KMap in various applications has been positive in terms of ease of use of the interface to create concept maps. Users across a range of disciplines have created extensive systems of concept maps and linked annotation related to projects and research topics. There is a wide range of individual variation in the perceived utility of concept maps, ranging from users who find them so natural that they will not undertake a project without first creating maps, to those who do not find such visual expression at all natural.

The more formal concept maps with well-defined meanings require extensive training in the semantics of the visual language to be used effectively. We have found, as Nosek and Roth (1990) have reported from human factors experiments, that users can comprehend knowledge structures in semantic networks more readily than in textual form. However, in creating maps, users tend to treat formal maps in the same way as informal ones, and generate visual knowledge representations that are meaningful to them but do not comply with the strict semantics of the formal language. Since users have similar problems with expression in the textual form of the specification language, the problem appears to be one of creating precise specifications rather than one of the use of visual languages.

The KMap system described in this article was designed to have an open architecture enabling it to emulate a wide range of concept map styles and applications, and to integrate readily with other applications. The examples given illustrate the flexibility and power of this approach. In future, as heterogeneous multi-part document standards such as OLE 2 (Microsoft, 1994) and OpenDoc (Apple, 1993b) become more widely used, it will be natural to provide KMap functionality through an embeddable component rather than a stand-alone application. This will enable concept maps to be embedded as active diagrams in documents being managed by other applications, such as World-Wide Web browsers.

It is proposed that the support of concept maps should be regarded as a fundamental requirement for any hypermedia system architecture. The maps have an abstract structure as sorted hypergraphs which makes it possible to treat them computationally as a very general data type from which application-specific types may be derived. There is a natural link from the type structure to the visual appearance which makes it possible to manage their human factors in a uniform and consistent way. There are also natural forms of direct manipulation of the visually presented maps that makes it possible to implement interaction with them through a few powerful primitives.

Acknowledgments

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. We are grateful to the anonymous referees for their detailed comments.

References

Apple (1993a). **Inside Macintosh Interapplication Communication**. Reading, Massachusetts, Addison-Wesley.

Apple (1993b). **OpenDoc Developer Overview**. Cupertino, California, Apple Corporation.

Ausubel, D.P. (1968). **Educational Psychology: A Cognitive View**. New York, Holt, Rinehart and Winston.

Axelrod, R. (1976). **Structure of Decision**. Princeton, New Jersey, Princeton University Press.

Baader, F. and Hollunder, B. (1991). KRIS: Knowledge representation and inference system. **ACM SIGART Bulletin 2**(3) 8-14.

Banathy, B.H. (1991). Cognitive mapping of educational systems for future generations. **World Futures 30**(1) 5-17.

Barrett, E., Ed. (1989). **The Society of Text: Hypertext, Hypermedia and the Social Construction of Information**. Cambridge, Massachusetts, MIT Press.

Barwise, J. and Etchemendy, J. (1990). Valid inference and visual representation. Zimmerman and Cunningham, Ed. **Visualization in Mathematics**. American Mathematical Association.

Berge, C. (1958). **The Theory of Graphs**. London, Methuen.

Berge, C. (1973). **Graphs and Hypergraphs**. London, North-Holland.

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H.F. and Secret, A. (1994). The World-Wide Web. **Communications ACM 37**(8) 76-82.

Bernstein, B. (1992). Euclid: Supporting collaborative argumentation with hypertext. Department of Computer Science, University of Colorado at Boulder. CU-CS-596-92.

Bertin, J. (1983). **Semiology of Graphics**. Madison, Wisconsin, University of Wisconsin Press.

Borgida, A., Brachman, R.J., McGuiness, D.L. and Resnick, L.A. (1989). CLASSIC: a structural data model for objects. **Proceedings of 1989 SIGMOD Conference on the Management of Data**. pp.58-67. New York, ACM Press.

Borgida, A. and Patel-Shneider, P.F. (1994). A semantics and complete algorithm for subsumption in the CLASSIC description logic. **Journal of Artificial Intelligence Research 1** 277-308.

Brachman, R.J. (1977). What's in a concept: structural foundations for semantic nets. **International Journal of Man-Machine Studies 9** 127-152.

Brachman, R.J. (1979). On the epistemological status of semantic nets. Findler, N.V., Ed. **Associative Networks: Representation and Use of Knowledge by Computers**. pp.3-50. New York, Academic Press.

Bradshaw, J.M., Ford, K.M., Adams-Webber, J.R. and Boose, J.H. (1993). Beyond the repertory grid: new approaches to constructivist knowledge acquisition tool development. **International Journal of Intelligent Systems 8**(2) 287-33.

Callon, M., Law, J. and Rip, A., Ed. (1986). **Mapping the Dynamics of Science and Technology**. Basingstoke, UK, MacMillan.

Cercone, N. and Schubert, L. (1975). Towards a state-based conceptual representation. **Proceedings of AAAI75**. pp.83-90. Los Altos, Morgan Kaufmann.

Chang, S.K., Ichikawa, T. and Ligomenides, P.A. (1986). **Visual Languages**. New York, Plenum Press.

CPCS (1993). **RepGrid 2 Manual**. Centre for Person Computer Studies, 3019 Underhill Drive NW, Calgary, AB, Canada T2N 4E4.

Cropper, S., Eden, C. and Ackermann, F. (1990). Keeping sense of accounts using computer-based cognitive maps. **Social Science Computer Review 8**(3) 345-366.

Eden, C., Jones, S. and Sims, D. (1979). **Thinking in Organizations**. London, Macmillan.

Ellis, G. and Levinson, R., Ed. (1992). **Proceedings of the First International Workshop on PEIRCE: A Conceptual Graphs Workbench**. University of Queensland, Australia, ftp.cs.uq.oz.au /pub /TECHREPORTS /department /TR0241.ps.Z.

Ellis, G., Levinson, R. and Robinson, P.J. (1994). Managing Complex Objects in PEIRCE. **International Journal Human-Computer Studies 4**(1/2) 109-148.

Fahlman, S.E. (1979). **NETL: A System for Representing and Using Real-World Knowledge**. Cambridge, Massachusetts, MIT Press.

Gaines, B.R. (1991). An interactive visual language for term subsumption visual languages. **IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence**. pp.817-823. San Mateo, California, Morgan Kaufmann.

Gaines, B.R. (1994a). Class library implementation of an open architecture knowledge support system. **International Journal Human-Computer Studies 41**(1/2) 59-107.

Gaines, B.R. (1994b). A situated classification solution of a resource allocation task represented in a visual language. **International Journal Human-Computer Studies 40**(2) 243-271.

Gaines, B.R. and Norrie, D.H. (1994). Mediator: information and knowledge management for the virtual factory. **SIGMAN AAAI-94 Workshop: Reasoning about the Shop Floor**. pp.30-39. Menlo Park, California, AAAI.

Gaines, B.R., Rappaport, A. and Shaw, M.L.G. (1992). Combining paradigms in knowledge engineering. **Data and Knowledge Engineering 9** 1-18.

Gaines, B.R. and Shaw, M.L.G. (1992a). Documents as expert systems. Bramer, M.A. and Milne, R.W., Ed. **Research and Development in Expert Systems IX. Proceedings of British Computer Society Expert Systems Conference**. pp.331-349. Cambridge, UK, Cambridge University Press.

Gaines, B.R. and Shaw, M.L.G. (1992b). Integrated knowledge acquisition architectures. **Journal for Intelligent Information Systems 1**(1) 9-34.

Gaines, B.R. and Shaw, M.L.G. (1993a). Basing knowledge acquisition tools in personal construct psychology. **Knowledge Engineering Review 8**(1) 49-85.

Gaines, B.R. and Shaw, M.L.G. (1993b). Open architecture multimedia documents. **Proceedings of ACM Multimedia 93**. pp.137-146.

Gaines, B.R. and Shaw, M.L.G. (1993c). Supporting the creativity cycle through visual languages. **AAAI Spring Symposium: AI and Creativity**. pp.155-162. Menlo Park, California, AAAI.

Gaines, B.R. and Shaw, M.L.G. (1994). Using knowledge acquisition and representation tools to support scientific communities. **AAAI'94: Proceedings of the Twelfth National Conference on Artificial Intelligence**. pp.707-714. Menlo Park, California, AAAI Press/MIT Press.

Golden, J.L., Berquist, G.F. and Coleman, W.E. (1976). **The Rhethoric of Western Thought**. Dubuque, Idaho, Kendall/Hunt.

Goodman, D. (1993). **The Complete AppleScript Handbook**. New York, Random House.

Graesser, A.C. and Clark, L.F. (1985). **Structures and Procedures of Implicit Knowledge**. New Jersey, Ablex.

Gui, J.-K. and Mäntylä, M. (1993). Assembly modeling on the basis of a mechanical design prototype. Laboratory for Information Processing Science, Helsinki University of Technology.

Hart, J.A. (1977). Cognitive maps of three latin american policy makers. **World Politics 30**(1) 115-140.

Jonassen, D.H. (1990). Semantic net elicitation: tools for structuring hypertext. McAleese, R. and Green, C., Ed. **Hypertext: State of the Art**. pp.142-152. Oxford, UK, Intellect.

Karnopp, D. and Rosenberg, R.C. (1975). **Systems Dyanmics: A Unified Approach**. New York, Wiley.

Karnopp, D., Rosenberg, R.C. and van Dixhorn, J.J. (1989). Bond Graph Techniques for Dynamic Systems in Engineering and Biology. **Journal Franklin Institute 308**(3)

Kelly, G.A. (1955). **The Psychology of Personal Constructs**. New York, Norton.

Kremer, R. and Gaines, B.R. (1994). Groupware concept mapping techniques. **Proceedings SIGDOC'94: ACM 12th Annual International Conference on Systems Documentation**. pp.156-165. New York, ACM.

Lambiotte, J.G., Dansereau, D.F., Cross, D.R. and Reynolds, S.B. (1989). Multirelational semantic maps. **Educational Psychology Review 1**(4) 331-367.

Mac Lane, S. (1971). **Categories for the Working Mathematician**. New York, Springer-Verlag.

Madan, M. (1994). Evaluating Alternative Approaches to Knowledge Acquisition for an Expert System. MSc. University of Calgary.

McKnight, C., Dillon, A. and Richardson, J. (1991). **Hypertext in Context**. Cambridge, UK, Cambridge University Press.

McNeese, M.D., Zaff, B.S., Peio, K.J., Snyder, D.E., Duncan, J.C. and McFarren, M.R. (1990). An Advanced Knowledge and Design Acquisition Methodology for the Pilot's Associate. Harry G Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio. AAMRL-TR-90-060.

Microsoft (1994). **OLE 2 Programmer's Reference Volume 1: Working with Windows Objects**. Redmond, Washington, Microsoft Press.

Moder, J.J. and Phillips, C.R. (1970). **Project Management with CPM and PERT`**. New York, Van Nostrand Reinhold.

Nersessian, N.J. (1989). Conceptual change in science and in science education. **Synthese 80**(1) 163-184.

Netscape (1995). Netscape Home Page. Netscape. http://www.netscape.com.

Nielsen, J. (1995). **Multimedia and Hypermedia: The Internet and Beyond**. Boston, Academic Press.

Nosek, J.T. and Roth, I. (1990). A comparison of formal knowledge representations as communication tools: predicate logic vs semantic network. **International Journal of Man-Machine Studies 33** 227-239.

Novak, J.D. (1977). **A Theory of Education**. Ithaca, Illinois, Cornell University Press.

Novak, J.D. and Gowin, D.B. (1984). **Learning How To Learn**. New York, Cambridge University Press.

Ousterhout, J.K. (1994). **Tcl and the Tk Toolkit**. Reading, Massachusetts, Adison-Wesley.

Quillian, M.R. (1968). Semantic memory. Minsky, M., Ed. **Semantic Information Processing**. pp.216-270. Cambridge, Massachusetts, MIT Press.

Reisig, W. (1985). **Petri Nets: An Introduction**. Berlin, Springer.

Roberts, D.D. (1973). **The Existential Graphs of Charles S. Peirce**. The Hague, Mouton.

Saint-Martin, F. (1990). **Semiotics of Visual Language**. Bloomington, Indiana University Press.

Shaw, M.L.G. and Gaines, B.R. (1992). Mapping creativity with knowledge support tools. **AAAI-91 Workshop on Creativity: Models, Methods and Tools**. pp.32-45. Menlo Park, California, AAAI.

Shaw, M.L.G., Gaines, B.R. and Linster, M. (1994). Supporting the knowledge engineering life cycle. Bramer, M.A. and Macintosh, A.L., Ed. **Research and Development in Expert Systems XI**. pp.73-86. Oxford, SGES Publications.

Shin, S.-J. (1994). **The Logical Status of Diagrams**. Cambridge, UK, Cambridge University Press.

Smolensky, P., Bell, B., Fox, B., King, R. and Lewis, C. (1987). Constraint-based hypertext for argumentation. **Proceedings of Hypertext'87**. pp.215-245. New York, ACM.

Sowa, J.F. (1984). **Conceptual Structures: Information Processing in Mind and Machine**. Reading, Massachusetts, Adison-Wesley.

Sowa, J.F., Ed. (1991a). **Principles of Semantic Networks: Explorations in the Representation of Knowledge**. San Mateo, California, Morgan-Kaufman.

Sowa, J.F. (1991b). Toward the expressive power of natural language. Sowa, J.F., Ed. **Principles of Semantic Networks: Explorations in the Representation of Knowledge**. pp.157-189. San Mateo, California, Morgan-Kaufman.

StarNine (1995). WebSTAR Home Page. StarNine. http://www.starnine.com/webstar/.

Streitz, N.A., Hannemann, J. and Thüring, M. (1989). From ideas and arguments to hyperdocuments: travelling through activity spaces. **Proceedings of Hypertext'89**. pp.343-364. New York, ACM.

Sun (1995). HotJava Home Page. Sun Microsystems. http://java.sun.com.

Thadgard, P. (1992). **Conceptual Revolutions**. Princeton, New Jersey, Princeton University Press.

Tomiyama, T. (1992). The technical concept of IMS. RACE Discussion Paper, No. RA-DP2, Research into Artifacts, Center for Engineering, The University of Tokyo.

Toulmin, S. (1958). **The Uses of Argument**. Cambridge, UK, Cambridge University Press.

Travers, M. (1989). A visual representation for knowledge structures. **Proceedings of Hypertext'89**. pp.147-158. New York, ACM.

Vickers, J.N. and Gaines, B.R. (1988). A comparison of books and hypermedia for knowledge-based sports coaching. **Microcomputers for Information Management 5**(1) 29-44.

Watanabe, H. (1989). Heuristic graph displayer for G-BASE. **International Journal of Man-Machine Studies 30**(3) 287-302.

Winer, D. (1992). **UserLand Frontier User Guide**. Palo Alto, California, UserLand Software.

Wittgenstein, L. (1953). **Philosophical Investigations**. Oxford, Blackwell.

Woods, W.A. (1975). What’s in a link: Foundations for semantic networks. Bobrow, D.G. and Collins, A.M., Ed. **Representation and Understanding: Studies in Cognitive Science**. pp.35-82. New York, Academic Press.

Woodward, B. (1990). Knowledge engineering at the front-end: defining the domain. **Knowledge Acquisition 2**(1) 73-94.