WEBMAP

CONCEPT MAPPING ON THE WEB

Brian R. Gaines, Mildred L. G. Shaw

Abstract

Concept maps have long provided visual languages widely used in many different disciplines and application domains. This article reports experience in taking an existing open architecture concept mapping tool and making it available on the Web in a number of ways: as a client helper downloading and uploading concept maps; as an active controller of the browser, indexing multimedia material through URLs embedded in concept maps; as a concept map creator controlled by the browser, generating concept maps through the browsing process; and as an auxiliary HTTP server making concept maps available as clickable maps for users who do not have the client helper or want to use active concept maps embedded in documents. ***Keywords****: Concept maps, semantic networks, active documents, client helpers, clickable map servers.*

Introduction

This article describes the integration of concept mapping tools with World Wide Web browsers and servers, as client helpers on the one hand and gatewayed, clickable image servers on the other. Abstractly, concept maps are sorted graphs visually represented as nodes having a type, name, and content, some of which are linked by arcs. Each type of node has associated visual attributes, such as shape and color scheme, and the arcs may be nondirectional, directional, or bidirectional. Links between nodes may also be labeled and typed by constructing them from a pair of arcs with an intervening node whose name is the link name and whose type is the link type. Figure 1 shows a typical concept map from a knowledge-based system that solves a room allocation problem [1].

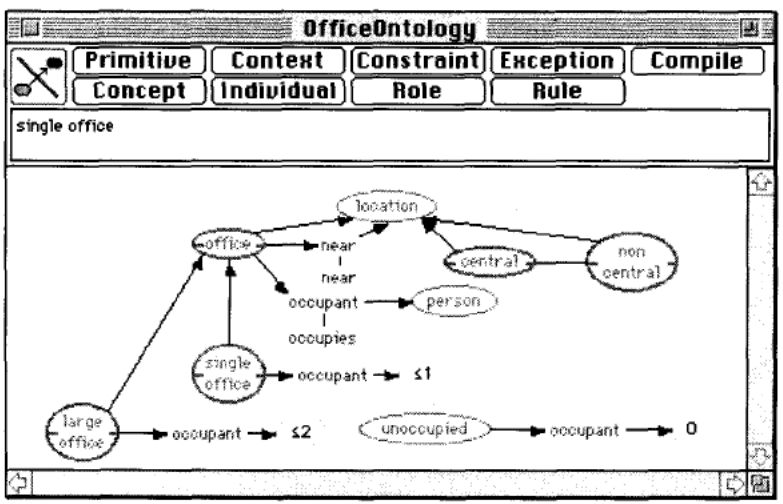

Figure 1: Concept map representing a semantic network

The map in Figure 1 is a *semantic network* [2] with a formal logical interpretation that allows it to be automatically compiled for a KL-ONE inference engine [3]. This is typical of artificial intelligence, engineering, and mathematical applications of concept maps. In these disciplines various forms of concept map are used as formal knowledge representation systems, for example, bond graphs in mechanical and electrical engineering [4], Petri nets in communications [5], and category graphs in mathematics [6]. More generally, concept maps have been used in education [7, 8], policy studies [9, 10], and the philosophy of science [11, 12] to provide a visual representation of knowledge structures and argument forms. They provide *visual languages* as complementary alternatives to natural languages as a means of communicating knowledge.

Concept maps have been used for a wide range of purposes, and it would be useful to make such usage available over the World Wide Web. This article reports on the development of WebMap, a system for integrating concept maps with existing Web media and technologies. WebMap is based on an existing open architecture concept mapping tool, KMap, developed for the Apple Macintosh platform. The next section gives an overview of KMap, and subsequent sections illustrate its application as a client helper to Web browsers and as an auxiliary server to Web servers.

KMap: A Concept Mapping Tool

KMap is a concept mapping tool written for Apple Macintosh computers. It is open architecture, first in allowing different forms of concept map to be defined and second in supporting integration with other applications through the Apple Object Event protocol and AppleScript. KMap has been used in a wide variety of situations ranging from concept mapping in education [13], through multimedia indexing [14] and semantic nets for knowledge-based systems [1, 15], to workflow support for scientific communities on the Internet [16-18].

Entering a Concept Map in KMap

Figure 2 shows a concept map from an application in which KMap was used to index a multimedia CD-ROM [19] of materials comprising the final report of the GNOSIS project [20], part of the international Intelligent Manufacturing Systems (IMS) research program [21]. The user creates a new node by selecting its type on the popup menu at the top left (currently showing type "ConMap" which is used to reference another concept map), typing its label into the text box below (currently showing "GNOSIS Final Reports"), and clicking on the popup menu text "ConMap." KMap then inserts a new node as shown, wrapping the text to create a horizontal rectangle.

The new node can be dragged anywhere in the window. The arrows to and from the new node are drawn by a simple process. The corners of the node are sensitive, and the cursor changes to an arrow symbol when over the lower right corner. Clicking on the mouse creates a "rubberband" line. Dragging this over another node highlights the node. Releasing the move over the highlighted node draws an arrow from the original node to the highlighted one.

Selecting Node Types in KMap

There are many different nodes types with differing appearances in the concept map of Figure 2. The selection of the node type is made through a popup menu. When the cursor browses over the node type at the top left it assumes a button shape, as shown in Figure 2, to indicate that clicking will create a new node. However, as the user moves to the right of the node type name, the cursor becomes a popup menu and mousing down produces the menu shown in Figure 3 which lists the available node types for selection. KMap also supports an alternative interface through buttons as shown at the top of Figure 1 where the node types are each listed on a separate button.

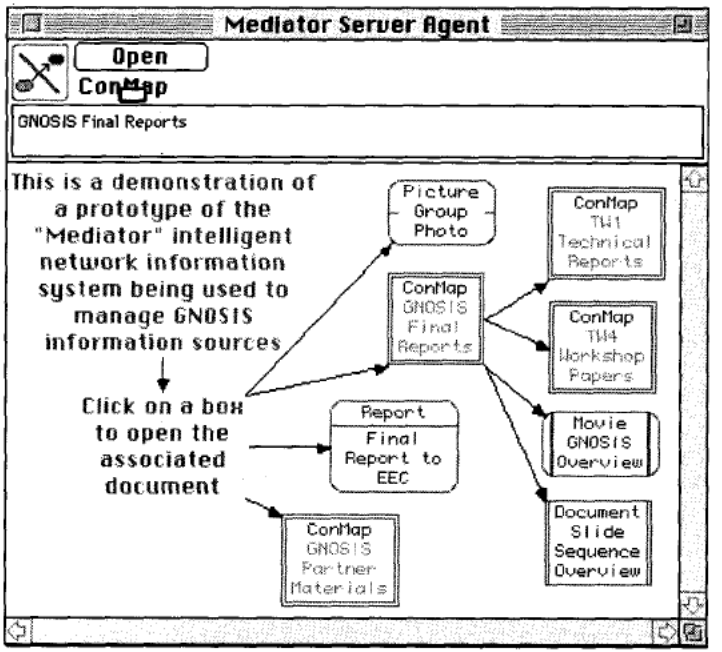

Figure 2: Concept map indexing a CD-ROM

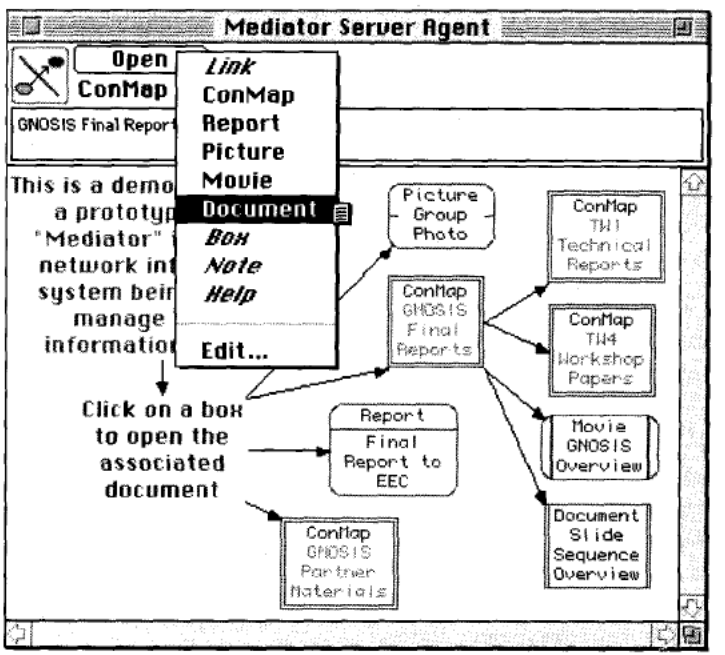

Figure 3: Popup menu for node types

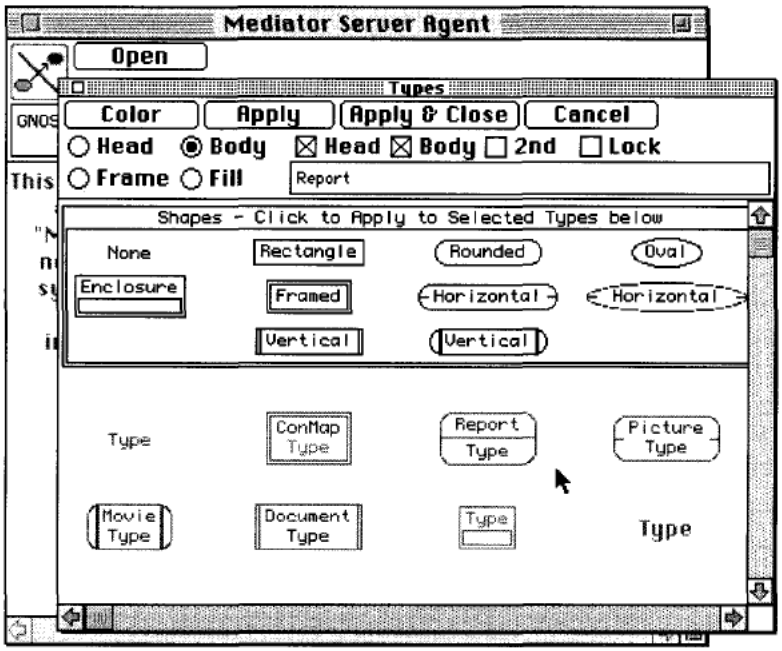

Figure 4: Node type editing and creation

Creating and Editing Node Types in KMap

At the bottom of the popup menu in Figure 3 there is an "Edit..." option. Selecting this opens a floating dialog as shown in Figure 4 which enables the user to edit existing node types and create new ones. The user creates a new node type by typing its name in the text box at the top, selecting a shape in the palette below, defining the colors of the surround, fill, heading text and body text, texts sizes and styles, and whether the head and/or body text should be shown.

Programming KMap

So far KMap has been shown as a flexible concept mapping or diagramming tool. However, it is also programmable so that user actions can communicate with other applications. Each node in a map has not only a label but also a hidden content, which is an arbitrary text string. The hidden content may be accessed and edited by holding down the "Option" key when double-clicking on a node. This brings up the content, rather than the label, in the text box for editing.

User actions in the concept map send messages to a script attached to the map which has access to all the map data, including the hidden content. This script is created and edited in a script editor activated by "Open Script" in the "File" menu as shown in Figure 5. The script receives messages when a concept map is open or closed, when it is edited in any way, when the user clicks in a node, a button, a popup menu, and so on. A full record of the map affected, the node within it, the menu selection, and so on, is passed as a parameter to the routine in the script. This enables concept maps to be used as active documents providing user interfaces to any functionality on the Macintosh, including control of other applications either locally or through a network. In particular, multiple versions of the same concept map open on different machines may be kept mutually updated as users edit them.

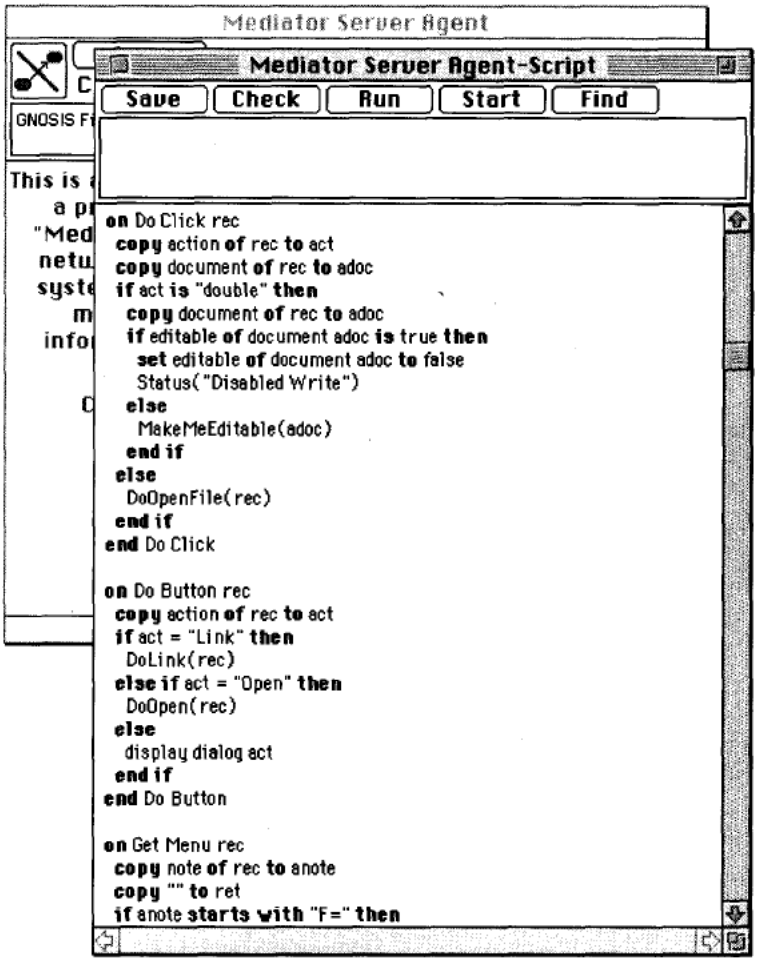

Figure 5: Scripting a concept map

Integrating KMap with Other Applications

KMap scripting has been used to create a wide range of computer-supported cooperative work applications in multimedia environments. The semantic network shown in Figure 1 can be sent to a compiler which converts it to the text form of KL-ONE definitions and loads these into the KRS [22] knowledge representation system for execution as part of the solution of a room allocation problem. The script for the concept map of Figure 2 provides multimedia indexing capabilities by interpreting a click on a node as a request to load the file specified by a URL stored in the node content. The file loaded can be another concept map, providing multi-level indexing, or it can be any file openable by another application. For example, the GNOSIS final report CD-ROM contains digitized photographs, digital movies, expert system knowledge bases, engineering drawings, reports in Word, Replica, and so on.

For example, in Figure 6 the node "GNOSIS Final Reports" has been clicked in the "Mediator Server

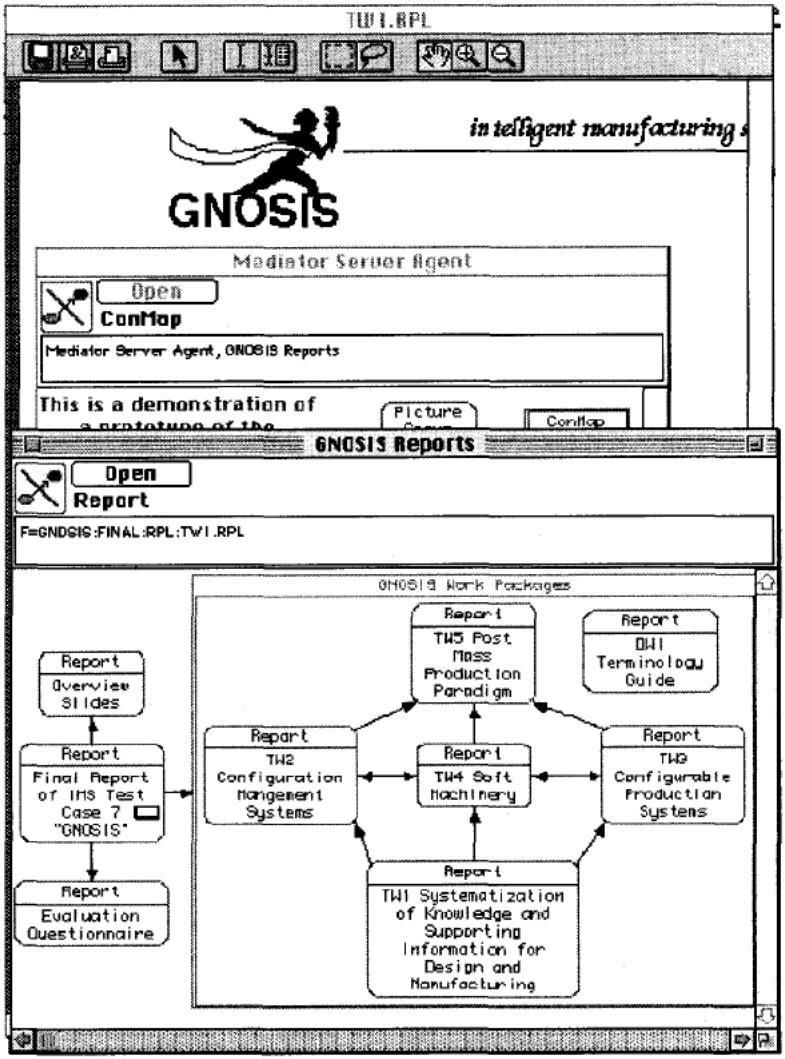

Figure 6: Multimedia indexing through concept maps

Agent" window of Figure 2. This has opened a concept map of the GNOSIS final reports as shown overlaying it in the lower part of the figure. Clicking on the node "Final Report of IMS Test Case 7 GNOSIS" at the middle left of this concept map has opened the Farallon Replica document shown in the window behind the two concept maps.

It should be noted that the only script operating is that in the "Mediator Server Agent" concept map. If it opens another concept map it links it to itself so that the script responds to actions in the new concept map, essentially acting as an active agent for otherwise passive documents. This is important to applications that fetch concept maps across the Internet since one does not want to accept arbitrary script functionality that can have detrimental effects on the local machine.

Using Concept Maps Across the World Wide Web

KMap is written in a C++ class library [23] that provides a wide range of application functionality including TCP/IP communications and protocols, typographic documents with active embedded components [24], knowledge processing, and so on. Hence, it has been simple to support collaborative groupware systems operating across local and wide area networks and providing end users with interfaces through concept maps and through active documents with embedded maps. However, much of the functionality required in these applications is now available with high quality at low cost in the form of World Wide Web browsers such as Mosaic and Netscape. These browsers support inter-application protocols allowing them to be integrated as components in a system involving other applications. They are also available on all major platforms making it possible to support a heterogeneous environment with minimal constraints on end users.

Hence, it is attractive to investigate the integration of the concept mapping techniques and tools already described with World Wide Web tools with a view to extending Web functionality to support collaborative activities with user interfaces largely through visual languages. In particular, the Mediator system illustrated above has been reimplemented to operate with HTTP servers and browsers, with KMap operating as a client helper tightly coupled to Netscape, and also with KMap operating as an auxiliary server either through its own HTTP support or through a common gateway interface (CGI) to WebStar. The following subsections illustrate some of the features of these implementations.

KMap as a Client Helper to Netscape

It was simple to reimplement the type of multimedia indexing functionality described above and illustrated in Figure 5. KMap was registered as a client helper with Netscape for the MIME type application/x-kss (which is the generic file type for all our knowledge support systems), and this MIME type was registered with our NCSA and WebStar servers and with Netscape as corresponding to the file suffix *.kss*. This is sufficient to allow concept maps to be fetched over the Web and opened in KMap. To allow the concept maps to be used as active documents to fetch other documents across the Web, a script was written for a concept map to act as a "Netscape Agent" similar to the "Mediator Server Agent" shown in Figure 4 but passing the appropriate part of the hidden content of nodes to Netscape using the "OpenURL" Apple event.

Figure 7 shows KMap acting as a client helper to Netscape to index the GNOSIS reports. The concept map of Figure 6 has been fetched as *GNOSISReports.kss*; clicking on the node "Final Report of IMS Test Case 7 GNOSIS" at the middle left has sent a message to Netscape requesting it open the URL *http://ksi.cpsc.ucalgary.ca/IMS/GNOSIS/TC7Final.kss*. This is another concept map shown at the bottom right which provides an overview of the structure of the GNOSIS final report. Clicking on the node "Project Objectives, Activities and Organization" has sent a message to Netscape requesting that it open the URL *http://ksi.cpsc.ucalgary.ca/IMS/GNOSIS/TC7FF2.kss*. This is an HTML document that Netscape opens in its own window as shown at the back. Clicking on other nodes in the concept map at the front will cause Netscape to index through the sections of the report.

Since the HTTP protocols support fetching documents in different formats for different applications, for example, fetching an AutoCAD dataset and opening it in AutoCAD, many of the design objectives of Mediator can be satisfied through this integration of KMap, Netscape, and the World Wide Web HTTP/HTML protocols. Mediator on the Web, coupled with various agent technologies, can provide a system supporting the manufacturing life cycle across a virtual enterprise distributed internationally [18].

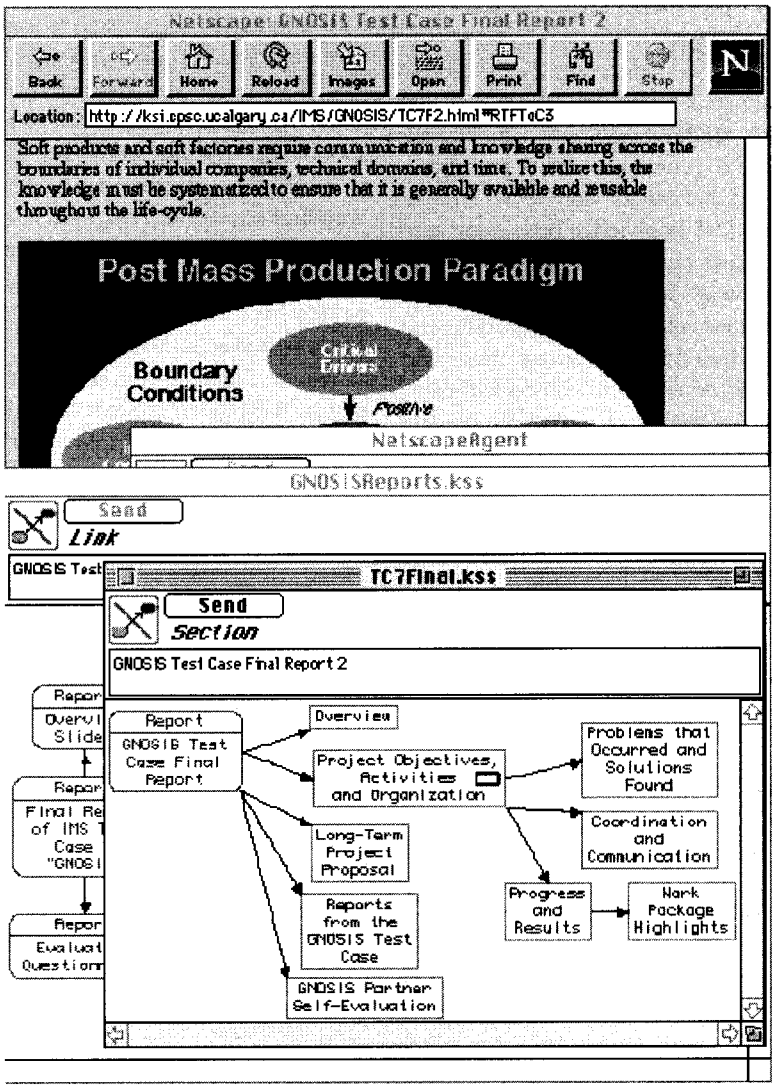

Figure 7: KMap as a client helper to Netscape

Generating Concept Maps by Browsing in Netscape

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 the World Wide Web, build a concept map of linked documents, and use this to provide access to those documents—a graphic "hotlist." Figure 8 shows such a hotlist being developed and used. The "NetscapeAgent" 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 sends 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 "NetscapeAgent" script stores the URLs and displays the name of the window in the text box near the top of its 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 hotlist of locations visited that can be used to access these locations again.

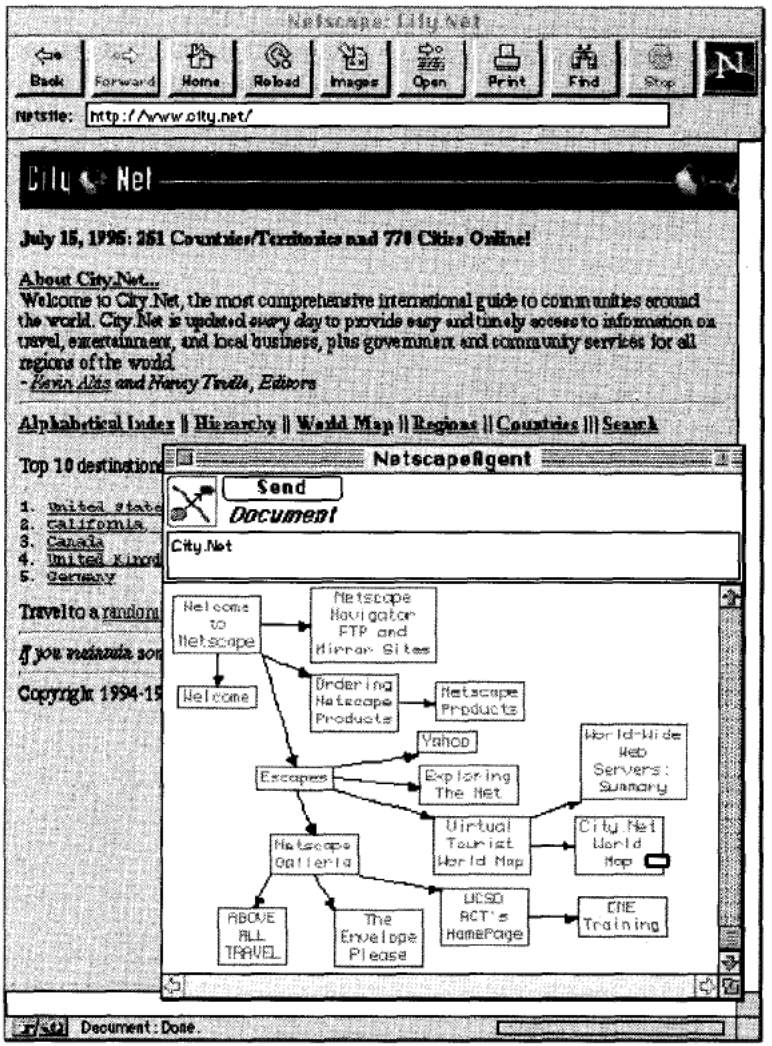

Figure 8: Concept maps being generated by Netscape

KMap as a Clickable Map Server to WebStar

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 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 HTTP server to allow the same concept maps to be used as clickable maps.

Clicking on the "Send" button at the top left of the "NetscapeAgent" concept map in Figure 8 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 WebStar.

KMap itself is also accessible through the CGI for two functions: first, to download concept maps as clickable maps in GIF format; and, second, to receive clicks on these maps and take the same action as if the concept map had been clicked locally in a KMap helper. Figure 9 shows the uploaded flow-chart concept map of Figure 8 being fetched using the URL ***http://tiger.cpsc.ucalgary.ca/WMClick/:X:NetscapeAgent.k*** which is recognized as an action to be passed to KMap by WebStar.

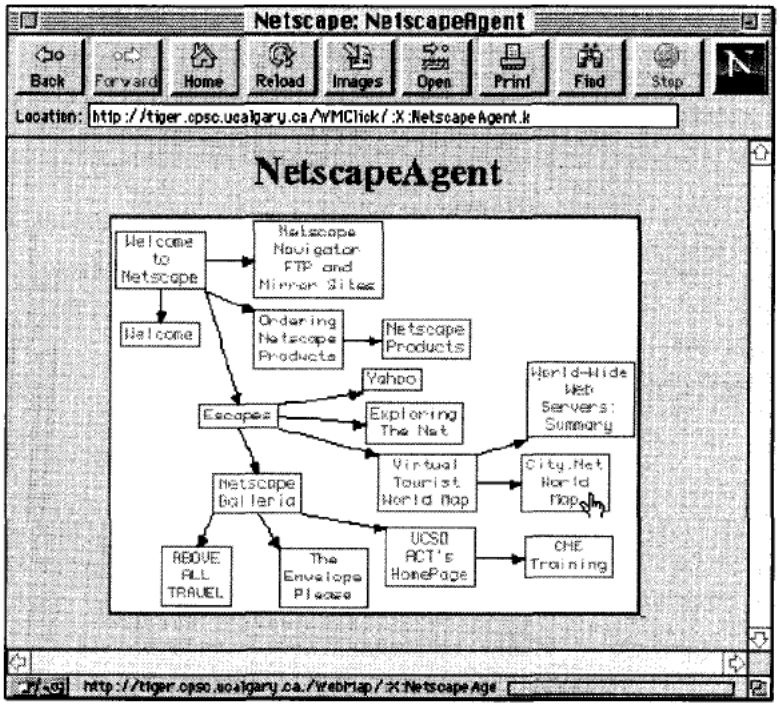

Figure 9: KMap acting as an HTTP server for clickable maps

KMap returns the HTML shown below 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 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 "City.Net World Map" 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 City.Net World Map.

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

Thus KMap supports the use of concept maps on 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 way 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 that any number of concept maps may be embedded as clickable maps in an HTML document. This enables much of the functionality of our active document tool, KWrite [24], 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 [15]. 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 [25].

Conclusion

Concept maps have long provided visual languages widely used in many different disciplines and application domains. Abstractly, they are sorted graphs visually represented as nodes having a type, name and content, some of which are linked by arcs. Concretely, they are structured diagrams having discipline- and domain-specific interpretations for their user communities and, sometimes, formally defining computer data structures. Concept maps have been used for a wide range of purposes, and it would be useful to make such usage available over the World Wide Web.

This article has reported experience in taking an existing open architecture concept mapping tool and making it available on the Web in a number of ways: as a client helper downloading and uploading concept maps through an open architecture browser such as Netscape; as an active controller of the browser, indexing multimedia material through URLs embedded in concept maps; as a concept map creator controlled by the browser, generating concept maps through the browsing process; and as an auxiliary HTTP server making concept maps available as clickable maps for users who do not have the client helper or want to use active concept maps embedded in documents. All of these various usages have been shown to complement one another effectively to support the simple and natural integration of concept maps with other World Wide Web media and technologies.

As the development of Web functionality proceeds it will become possible to embed concept mapping tools even more effectively as active components of HTML documents. Meanwhile the

current development of WebMap shows that much can be achieved within the existing Web capabilities. ■

Acknowledgments

This work was funded in part by the Natural Sciences and Engineering Research Council of Canada.

References

1. Gaines, B.R., *A situated classification solution of a resource allocation task represented in a visual language. International Journal Human-Computer Studies* 40(2), 1994, pp. 243-271.
2. Sowa, J.F., ed. *Principles of Semantic Networks: Explorations in the Representation of Knowledge.* San Mateo, California: Morgan-Kaufman, 1991.
3. Gaines, B.R., *An interactive visual language for term subsumption visual languages,* in *IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence.* San Mateo, California: Morgan-Kaufman, 1991, pp. 817-823.
4. Karnopp, D., R.C. Rosenberg, and J.J. van Dixhorn, *Bond Graph Techniques for Dynamic Systems in Engineering and Biology. Journal Franklin Institute* 308(3), 1989.
5. Reisig, W., *Petri Nets: An Introduction.* Berlin: Springer, 1985.
6. Mac Lane, S., *Categories for the Working Mathematician.* New York: Springer-Verlag, 1971.
7. Lambiotte, J.G., et al., *Multirelational semantic maps.* Educational Psychology Review 1(4), pp. 331-367, 1989.
8. Novak, J.D., and D.B. Gowin, *Learning How To Learn,* New York: Cambridge University Press, 1984.
9. Axelrod, R., *Structure of Decision,* Princeton, New Jersey: Princeton University Press, 1976.
10. Eden, C., S. Jones, and D. Sims, *Thinking in Organizations,* London: Macmillan, 1979.
11. Toulmin, S., *The Uses of Argument,* Cambridge, UK: Cambridge University Press, 1958.
12. Thadgard, P., *Conceptual Revolutions,* Princeton, New Jersey: Princeton University Press, 1992.
13. Gaines, B.R., and M.L.G. Shaw, *Supporting the creativity cycle through visual languages,* in *AAAI Spring Symposium: AI and Creativity,* Menlo Park, California: AAAI, 1993, pp. 155-162.
14. Gaines, B.R., and M.L.G. Shaw, *Concept maps indexing multimedia knowledge bases,* in *AAAI-94 Workshop: Indexing and Reuse in Multimedia Systems,* Menlo Park, California: AAAI, 1994, pp. 36-45.
15. Gaines, B.R., and M.L.G. Shaw, *Documents as expert systems,* in *Research and Development in Expert Systems IX. Proceedings of British Computer Society Expert Systems Conference,* M.A. Bramer and R.W. Milne, Editors, Cambridge, UK: Cambridge University Press, 1992, pp. 331-349.
16. Gaines, B.R., and M.L.G. Shaw, *Using knowledge acquisition and representation tools to support scientific communities,* in *AAAI'94: Proceedings of the Twelfth National Conference on Artificial Intelligence,* Menlo Park, California: AAAI Press/MIT Press, 1994, pp. 707-714.
17. Gaines, B.R., and D.H. Norrie, *Mediator: information and knowledge management for the virtual factory,* in *SIGMAN AAAI-94 Workshop: Reasoning about the Shop Floor,* Menlo Park, California: AAAI, 1994, pp. 30-39.
18. Gaines, B.R., D.H. Norrie, and A.Z. Lapsley, *Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise,* in *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics,* New York: IEEE, 1995.
19. *GNOSIS: Intelligent Manufacturing Systems: IMS Test Case 7: Hybrid-CD, Macintosh (native), PC, Unix (ISO 9660).* CD-GNOSIS-May 94, Calgary, Canada: Knowledge Science Institute and Division of Manufacturing Engineering, University of Calgary, 1994.
20. *Knowledge Systematization: Configuration Systems for Design and Manufacturing: Final Report of the Test Case. http://ksi.cpsc.ucalgary.ca/IMS/GNOSIS,* Calgary, Canada: Knowledge Science Institute, University of Calgary, 1994.
21. Gaines, B.R., and D.H. Norrie, *Knowledge systematization in the international IMS research program,* in *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics,* New York: IEEE, 1995.
22. Gaines, B.R., *Empirical investigations of knowledge representation servers: Design issues and applications experience with KRS. ACM SIGART Bulletin* 2(3), 1991, pp. 45-56.
23. Gaines, B.R., *Class library implementation of an open architecture knowledge support system. International Journal Human-Computer Studies,* 41(1/2), 1994, pp. 59-107.

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

25. Sun, *HotJava Home Page. http://java.sun.com*, Sun Microsystems, 1995.

About the Authors

Brian R. Gaines
[http://ksi.cpsc.ucalgary.ca/KSI]
Knowledge Science Institute
University of Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca

Dr. Brian R Gaines is Killam Memorial Research Professor and Director of the Knowledge Science Institute at the University of Calgary. His previous positions include Professor of Industrial Engineering at the University of Toronto, Technical Director and Deputy Chairman of the Monotype Corporation, and Chairman of the Department of Electrical Engineering Science at the University of Essex. He received his BA, MA, and PhD from Trinity College, Cambridge, and is a Chartered Engineer, Chartered Psychologist, and a Fellow of the Institution of Electrical Engineers, the British Computer Society and the British Psychological Society. He is editor of the *International Journal of Human-Computer Studies* and *Knowledge Acquisition*, and of the *Computers and People* and *Knowledge-Based Systems* book series. He has authored over 400 papers and authored or edited 10 books on a wide variety of aspects of computer and human systems. His research interests include the socio-economic dynamics of science and technology; the nature, acquisition and transfer of knowledge; software engineering for heterogeneous systems; knowledge-based system applications in manufacturing, the professions, sciences and humanities; and the systematic acceleration of scientific research through computer-mediated collaboration.

Mildred L. G. Shaw
[http://ksi.cpsc.ucalgary.ca/KSI]
Knowledge Science Institute
University of Calgary, Alberta, Canada T2N 1N4
mildred@cpsc.ucalgary.ca

Dr. Mildred L G Shaw is Professor of Computer Science at the University of Calgary. She received her BSc and MSc from the University of London, and her PhD from Brunel University, and is a Chartered Mathematician and Chartered Psychologist. She is a Fellow of the Institute of Mathematics and its Applications and the British Computer Society and an Associate Fellow of the British Psychological Society. Dr. Shaw is a member of the editorial boards of the *International Journal of Human-Computer Studies* and *Applied Intelligence*. She has authored over 150 papers and authored or edited 5 books on a wide variety of aspects of computer and human systems. Her research interests include human-computer interaction; the acquisition and transfer of knowledge; software engineering; and knowledge-based system applications.