Implementing the Learning Web

Brian R. Gaines and Mildred L. G. Shaw
University of Calgary
gaines@cpsc.ucalgary.ca, mildred@cpsc.ucalgary.ca

Abstract

The implementation of a system to support a learning web (Norrie and Gaines, 1995) involves the coordinated support of a wide range of knowledge processes in a learning community. These include the discourse processes among members of the community, the dissemination of knowledge through documents and multimedia resources, the elicitation, sharing and comparison of conceptual structures, and the exploration of the consequences of knowledge through simulation and application. Currently the Internet provides excellent tools supporting discourse such as email list servers, and supporting multimedia resource dissemination through World Wide Web (web) servers and browsers. Mediator, a prototype learning web implementation, involves integrating these existing tools with new ones supporting the collaborative modeling of conceptual structures through questionnaires, concept maps and repertory grids. This article describes the Mediator architecture and its implementation on the web, and describes and exemplifies the use of existing facilities and the additional tools.

1 Introduction

The design of technologies to support human knowledge processes in a learning web (Norrie and Gaines, 1995) needs itself to be evolutionary, providing not only tools to support learning but also the capability for the web technology itself to change and grow as the web changes and grows. It would be inconsistent with the basic principles of the learning web to design a monolithic system to support the learning web in all respects for all time. Any design has to have an open architecture that is able to assimilate or accommodate technologies from many sources, is never assumed to be complete, and may have to undergo radical change as needs, and our understanding of them, changes with time and in response to the initial implementations of the web. The logic of the web also dictates that no one group, institution, organization or even nation, should assume it can develop all the necessary learning web technologies. The development of the web is intrinsically a somewhat anarchic collaborative, perhaps best modeled on the growth of the Internet to date.

This article discusses the overall design for a learning web and then describes and illustrates some of the major technologies developed so far.

2 Mediator Architecture for the Learning Web

Norrie and Gaines (1995) use the term Mediator to describe the general architecture of information technology to support the learning web since the requirement can be modeled as one of mediating between learning agents. Figure 1 shows the design of the Mediator architecture as a four-layer distributed client, distributed server design, in which multiple users can collaborate synchronously or asynchronously through processes running anywhere on the network. A heterogeneous environment is assumed in which there are multiple protocols and multiple forms of user interface. The collaborating and geographically dispersed user community interacts with the system through a variety of forms of user interface, typically graphics, text, visual languages and hypermedia. They access a variety of ‘applications’ defined as packages of functionality providing defined services. The functionality is made operational by initiating processes which may run anywhere on the network; that is, remote procedure calls are expected to be common. Inter-process communication is provided through local and wide area networks supporting a range of knowledge and data interchange formats.

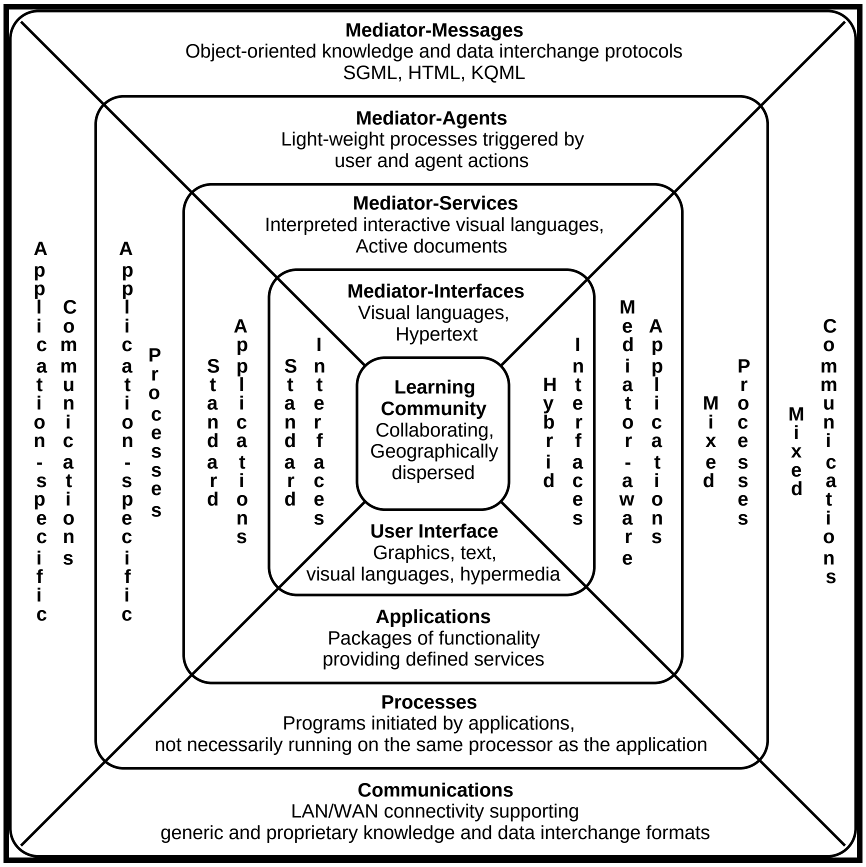


Figure 1 Mediator architecture supporting the learning web

On the left of Figure 1, the four layers are shown instantiated in terms of conventional software packages designed independently of Mediator: as standard interfaces, standard applications, application-specific processes and application-specific communications. Such standard applications can play a role in a Mediator-coordinated system, minimally by the application and its datasets being registered in Mediator with its application windows open on the Mediator desktop, and maximally by Mediator controlling its inputs, outputs and operations by job-control scripts. On the right, the four layers are shown instantiated in terms of separately designed applications that are ‘Mediator-aware’ to some extent, for example in using the Mediator interface technology, applications, agents or protocols as part of their normal operation. The Mediator shell technology is being designed to be highly modular and readily integrated in whole or in part with existing applications.

3 Implementing Mediator on the Internet and World Wide Web

The original implementation of prototype Mediator systems for manufacturing applications involved the complete development of special-purpose software (Gaines and Norrie, 1994; Maturana and Norrie, 1995), a massive task given the scope of the system. Recently, we have experimented with realizing much of the functionality through extension to the World Wide Web (Gaines, Norrie and Lapsley, 1995), and in extending Mediator for learning web applications we have increasingly based the design on the integration of existing Internet technologies, notably email list servers and web servers and browsers. This has enabled us to implement major subsystems very rapidly and bring them into use experimentally in existing undergraduate courses. In particular we have been able to integrate our existing concept mapping tools (Gaines and Shaw, 1993b; Gaines and Shaw, 1995b) in such a way that they can be used as both web browser helpers and auxiliary web servers (Gaines and Shaw, 1995a), and to port our existing repertory grid conceptual modeling tools (Shaw and Gaines, 1992; Gaines and Shaw, 1993a) to operate completely through the web as auxiliary web servers (Shaw and Gaines, 1995).

3.1 Supporting Human Discourse through the Internet and World Wide Web

The base level technologies for implementing the learning web are those for supporting human discourse. Computer agents have no role to play until the knowledge processes of communication and coordination among human agents are supported. These human processes are basically those of discourse through electronic mail, emulating human conversation, and discourse through multimedia documents, emulating the role of media such as papers, books and videos. Media enable knowledge to be captured as what Popper (1968) terms “world 3” objects, the expressed products of the human mind that continue to exist independently of their originators. Email list servers support group interaction at a conversational level, and web browsers support group interaction at a media level.

These two capabilities may be combined as shown on the left hand side of Figure 2. An integrated email/web client such as Netscape 2.0 can be used to support conversational discourse through a list server, and media preparation and access through a web server. In addition, as shown at the bottom left, the conversational discourse can be archived and automatically converted to hypertext files accessible through the web. Figure 3 shows the archives of a list server converted to a web document and indexed by topic.

The basic list and web servers, coupled through hypermail, provide excellent support of discourse in a learning environment, and are becoming increasingly used at all levels of education. The learning web implementation builds on this base level in a consistent fashion by providing special-purpose client helpers that extend the functionality of the web browser, and special-purpose auxiliary servers that extend the functionality of the web server. The technology to do this is described in the next section.

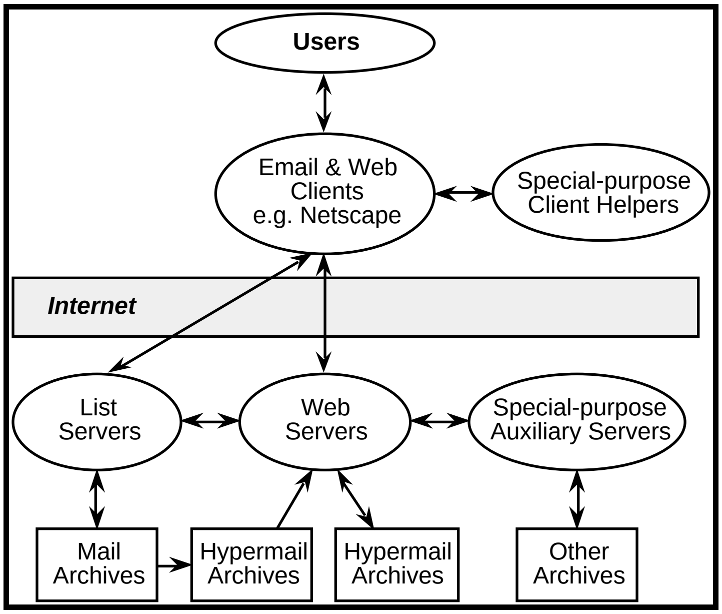
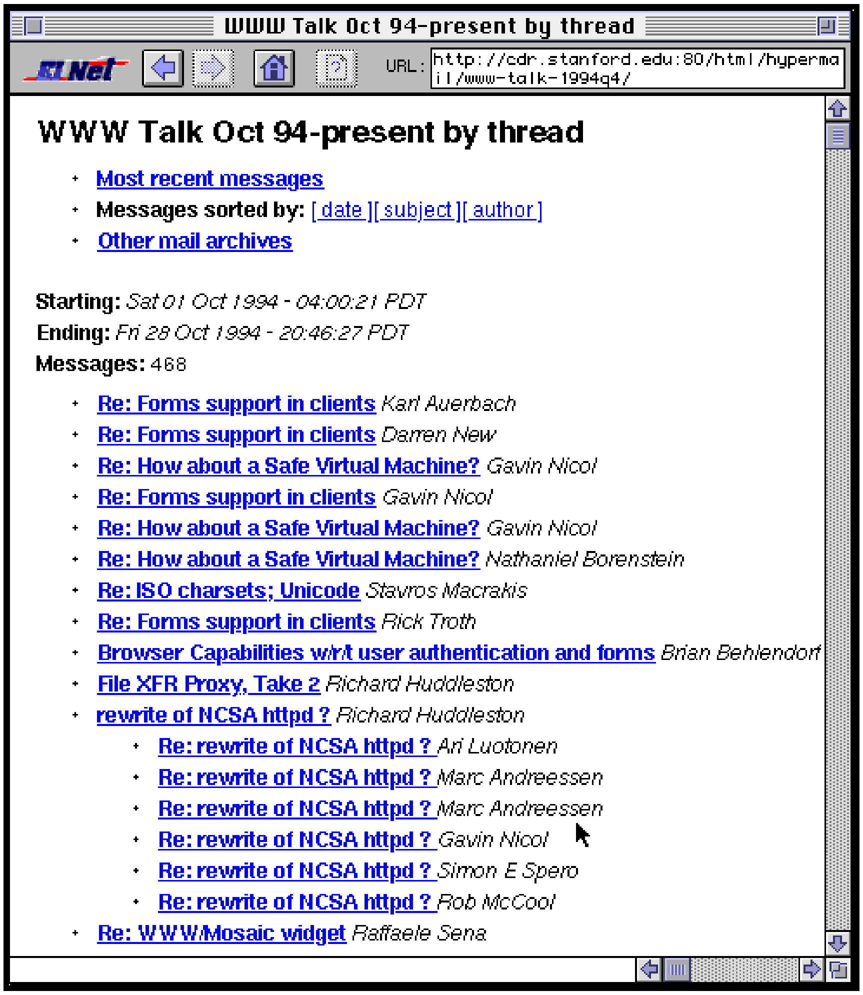


Figure 2 Internet/Web implementation of Mediator


WWW Talk Oct 94-present by thread
EINet
URL: http://cdr.stanford.edu:80/html/hypermail/www-talk-1994q4/

WWW Talk Oct 94-present by thread

- Most recent messages
- Messages sorted by: [date][subject][author]
- Other mail archives

Starting: *Sat 01 Oct 1994 - 04:00:21 PDT*
Ending: *Fri 28 Oct 1994 - 20:46:27 PDT*
Messages: 468

- Re: Forms support in clients *Karl Auerbach*
- Re: Forms support in clients *Darren New*
- Re: How about a Safe Virtual Machine? *Gavin Nicol*
- Re: Forms support in clients *Gavin Nicol*
- Re: How about a Safe Virtual Machine? *Gavin Nicol*
- Re: How about a Safe Virtual Machine? *Nathaniel Borenstein*
- Re: ISO charsets; Unicode *Stavros Macrakis*
- Re: Forms support in clients *Rick Troth*
- Browser Capabilities w/r/t user authentication and forms *Brian Behlendorf*
- File XFR Proxy, Take 2 *Richard Huddleston*
- rewrite of NCSA httpd ? *Richard Huddleston*
  - Re: rewrite of NCSA httpd ? *Ari Luotonen*
  - Re: rewrite of NCSA httpd ? *Marc Andreessen*
  - Re: rewrite of NCSA httpd ? *Marc Andreessen*
  - Re: rewrite of NCSA httpd ? *Gavin Nicol*
  - Re: rewrite of NCSA httpd ? *Simon E Spero*
  - Re: rewrite of NCSA httpd ? *Rob McCool*
- Re: WWW/Mosaic widget *Raffaele Sena*

Figure 3 Archives of a list server converted and accessed through the web

3.2 Forms as Programmable Graphic User Interfaces Embedded in Web Documents

In March 1993 the web was still being presented as primarily a hypermedia retrieval system (Berners-Lee, 1993), but in November that year a development took place that so changed the nature of the web as to constitute a revolution in its own right. Andreessen (1993) issued NCSA Mosaic version 2 using extended HTML (hypertext markup language) tags to encode definitions of Motif widgets embedded as *forms* within a hypermedia document, and allowed the state of those widgets within the client to be transmitted to the server. Suddenly the web protocols transcended their original conception to become the basis of general interactive, distributed, client-server information systems.

What changed with the advent of the forms capability was that the client in Figure 2 became able to transmit structured information from the user back to an arbitrary application gatewayed through the server. The server could then process that information and generate an HTML document which it sent back as a reply. This document could itself contain forms for further interaction with the user, thus supporting a sequence of client-server transactions.

In essence, the HTML tagged encoding schema allows documents to include not only text, typographic and multimedia material but also to carry arbitrary additional data through simple, backwards compatible extensions. The definition of HTML is currently being standardized at four levels (Berners-Lee, Connolly and Muldrow, 1994):

level 0 - text with embedded links;

level 1 - adds typographic text with embedded images;

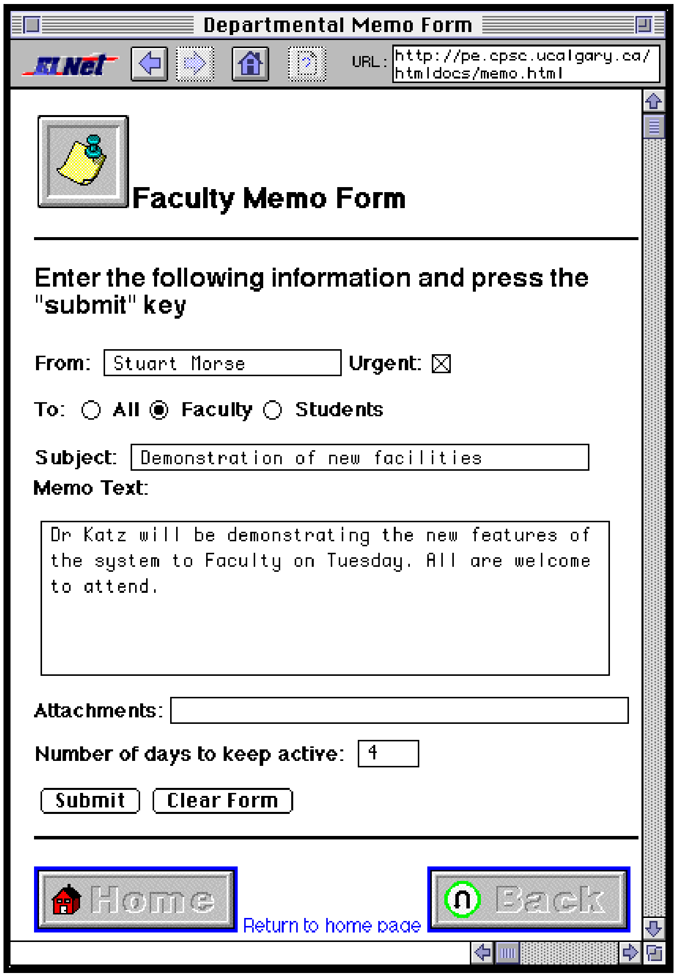
level 2 - adds embedded graphic user interfaces (forms);

level 3 - adds tables, mathematics, and more.

Level 0 functionality can be supported on an alphanumeric terminal. Level 1 adds typography and pictures. Level 2 adds embedded graphic user interfaces. Level 3 is still somewhat open-ended and will evolve through prototype implementations (Raggett, 1994).

The level 2 forms extension enhances the capability of HTML documents to act as graphic user interfaces by allowing other widgets such as buttons, check boxes, radio buttons, popup menus, scrolling lists, and text entry boxes to be embedded. Figure 4 shows a memo entry facility implemented for our Faculty of Kinesiology implemented using an HTML form. The user can type information into what appears to be a normal dialog box, and submit that information to the server.

The client-server capabilities of the web have been used to develop additional tools that enhance the capabilities of the web to support collaborative conceptual modeling which makes knowledge and knowledge processes explicit as part of the learning web implementation. Sub-systems for repertory grid elicitation and conceptual mapping are described in the following sections.



Departmental Memo Form

URL: http://pe.cpsc.ucalgary.ca/htmldocs/memo.html

Faculty Memo Form

Enter the following information and press the "submit" key

From: Stuart Morse Urgent: ☒

To: ○ All ● Faculty ○ Students

Subject: Demonstration of new facilities

Memo Text:

Dr Katz will be demonstrating the new features of the system to Faculty on Tuesday. All are welcome to attend.

Attachments:

Number of days to keep active: 4

Submit Clear Form

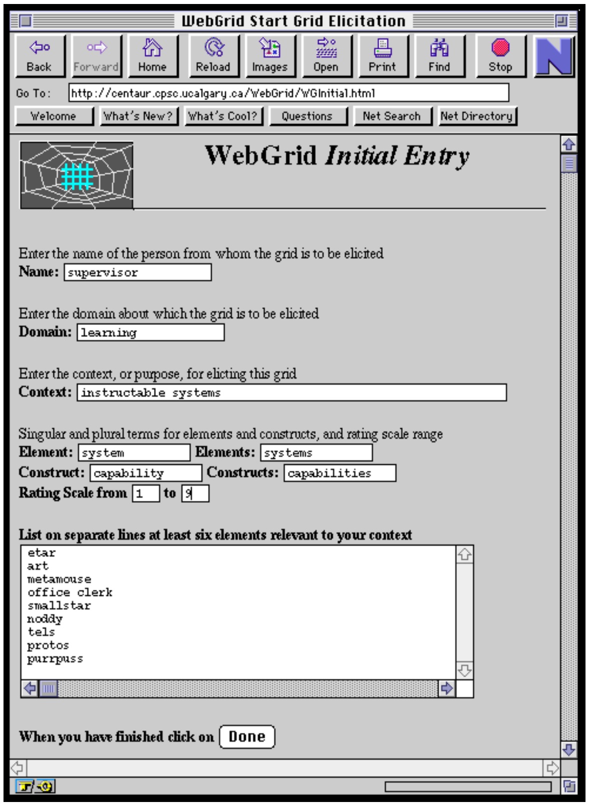
Return to home page

Figure 4 Graphic user interface embedded in a web document

4 Repertory Grid Conceptual Modeling on the Web

The repertory grid is a tool based on personal construct psychology (Kelly, 1955) to model conceptual structures by eliciting relevant distinctions about particular cases. An interactive computer program elicits the significant distinctions between cases all the time feeding back matches between cases to elicit new distinctions, and matches between distinctions to elicit new cases (Gaines and Shaw, 1993a). The resultant grid is clustered to feed back to the individual the overall conceptual structure for validation, and grids from different individuals may be compared to present the similarities and differences in their conceptual structures (Gaines and Shaw, 1989).

WebGrid (Shaw and Gaines, 1995) is an implementation of a repertory grid elicitation program as auxiliary server on the web, using HTML forms to support an elicitation dialog with the user. Figure 5 shows the supervisor of an MSc student using WebGrid through Netscape to start a repertory grid elicitation in the research domain of his student, “learning,” that is specifically concerned with the supervisor’s and student’s understanding of “instructable systems.” The supervisor has entered a list of 9 elements, in this case concrete examples of instructable systems, on which to base the elicitation of the way in which he construes them.



WebGrid *Initial Entry*

Enter the name of the person from whom the grid is to be elicited
Name: supervisor

Enter the domain about which the grid is to be elicited
Domain: learning

Enter the context, or purpose, for elicting this grid
Context: instructable systems

Singular and plural terms for elements and constructs, and rating scale range
Element: system **Elements:** systems
Construct: capability **Constructs:** capabilities
Rating Scale from 1 **to** 9

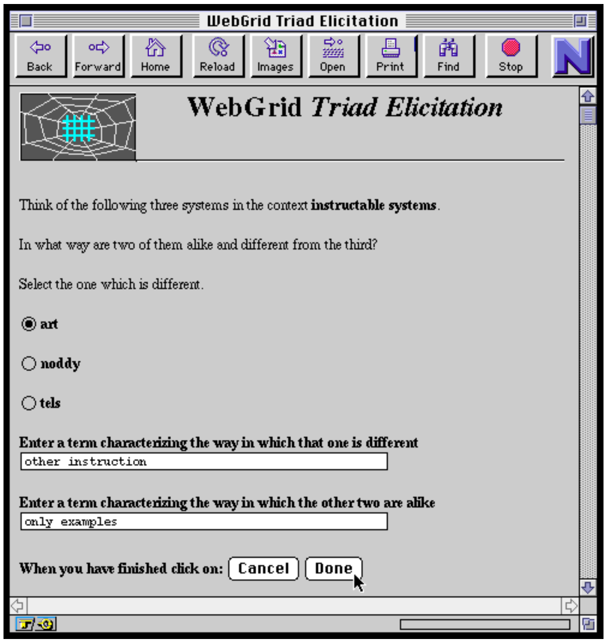
List on separate lines at least six elements relevant to your context

```
etar
art
metamouse
office clerk
smallstar
noddy
tels
protos
purrpuss
```

When you have finished click on Done

Figure 5 WebGrid initial entry screen

When the supervisor has entered the data shown in Figure 5, he clicks on "Done" and WebGrid generates the screen in Figure 6 where he is asked to distinguish three of his elements using the standard repertory grid triadic construct elicitation methodology.



WebGrid *Triad Elicitation*

Think of the following three systems in the context **instructable systems**.

In what way are two of them alike and different from the third?

Select the one which is different.

● **art**

○ **noddy**

○ **tels**

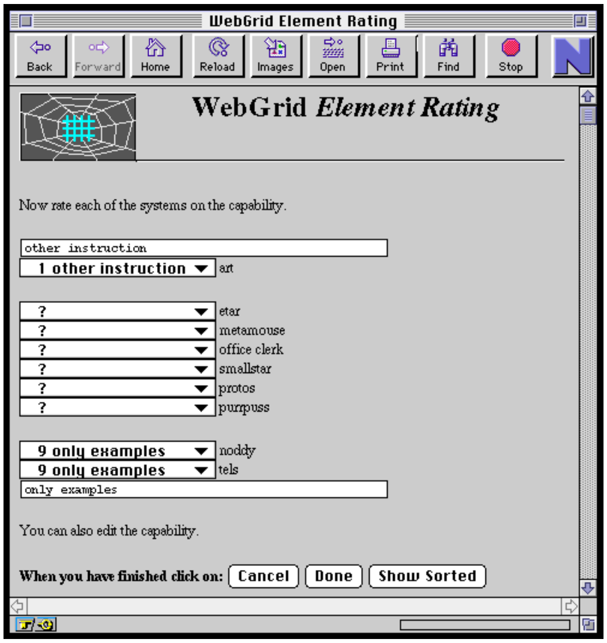
Enter a term characterizing the way in which that one is different
other instruction

Enter a term characterizing the way in which the other two are alike
only examples

When you have finished click on: Cancel Done

Figure 6 WebGrid elicitation of a construct from a triad of elements

He clicks on “Done” and WebGrid generates HTML for the screen in Figure 7 where the elements are shown alongside popup menus which can be used to rate them on the new construct as shown in Figure 8.



WebGrid *Element Rating*

Now rate each of the systems on the capability.

other instruction
1 other instruction ▼ art

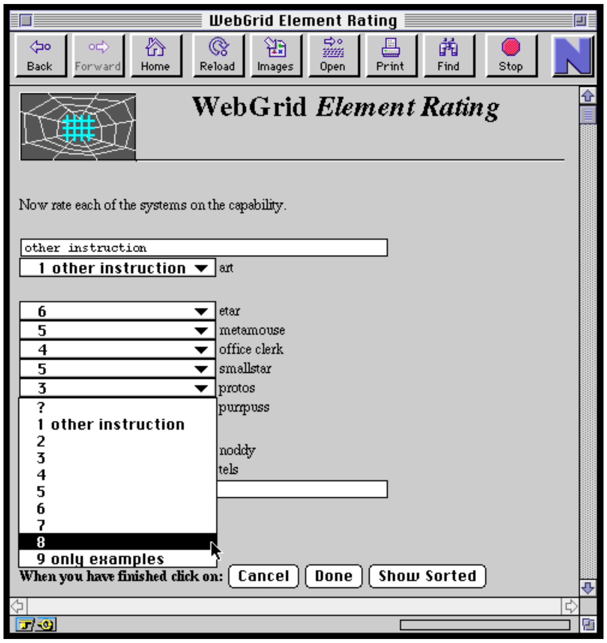
? ▼ etar
? ▼ metamouse
? ▼ office clerk
? ▼ smallstar
? ▼ protos
? ▼ purrpuss

9 only examples ▼ noddy
9 only examples ▼ tels
only examples

You can also edit the capability.

When you have finished click on: Cancel Done Show Sorted

Figure 7 WebGrid rating of elements on a new construct



WebGrid *Element Rating*

Now rate each of the systems on the capability.

other instruction
1 other instruction ▼ art

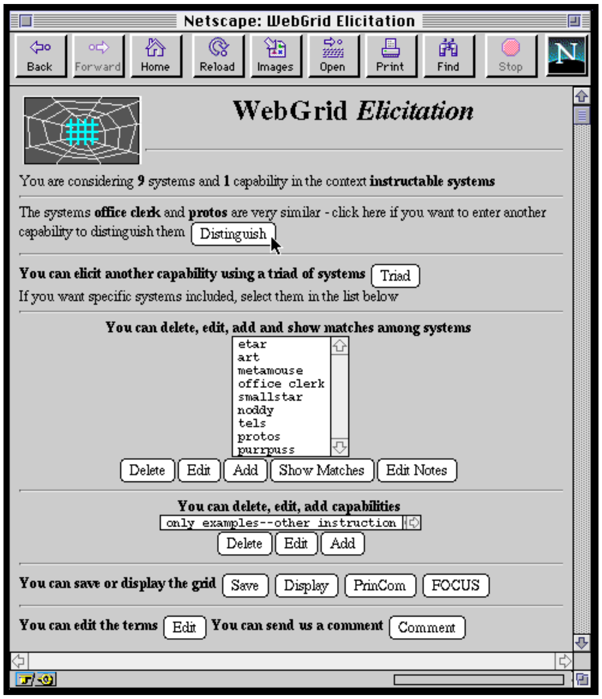
6 ▼ etar
5 ▼ metamouse
4 ▼ office clerk
5 ▼ smallstar
3 ▼ protos
purrpuss

noddy
tels

When you have finished click on: Cancel Done Show Sorted

Figure 8 WebGrid rating of elements on a new construct—popup menu scales

When the supervisor has rated each element on the new construct, he clicks on “Done” and WebGrid generates HTML for the screen shown in Figure 9 which shows the elements and constructs entered so far, and the various options available to the user. These allow the grid to be examined, edited, analyzed, and so on.



WebGrid *Elicitation*

You are considering **9** systems and **1** capability in the context **instructable systems**

The systems **office clerk** and **protos** are very similar - click here if you want to enter another capability to distinguish them Distinguish

You can elicit another capability using a triad of systems Triad
If you want specific systems included, select them in the list below

You can delete, edit, add and show matches among systems

```
etar
art
metamouse
office clerk
smallstar
noddy
tels
protos
purrpuss
```

You can delete, edit, add capabilities

```
only examples--other instruction
```

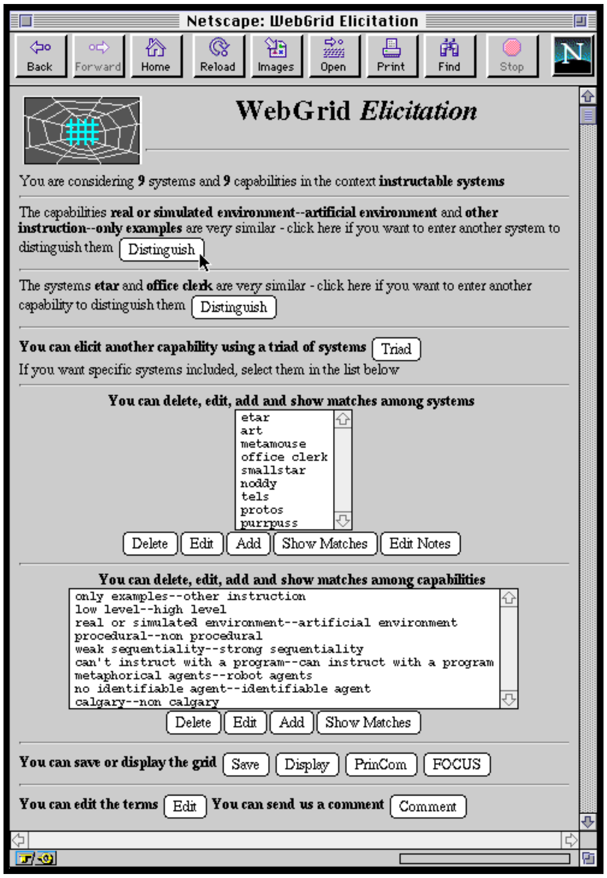
You can save or display the grid Save Display PrinCom FOCUS

You can edit the terms Edit **You can send us a comment** Comment

Figure 9 WebGrid main display of elements, constructs and functionality

This main screen is generated in sections, each of which give the user different information relevant to the elicitation. For example, the first suggestion is that a new construct be added to distinguish between the elements, “office clerk” and “protos.” If the user clicks on “Distinguish” WebGrid will generate HTML for a screen to enter a construct with “office clerk” at one pole and “protos” at the other. When the construct has been entered and the user clicks on “Done”, WebGrid will generate a screen for rating all the new elements on the new construct similar to that of Figure 7.

Figure 10 shows the main screen when the user has entered 9 constructs. Now construct matches are apparent, and the option at the top of the screen is to enter a new element to break the match between the constructs “weak sequentiality—strong sequentiality” and “non procedural—procedural.”



WebGrid *Elicitation*

You are considering **9** systems and **9** capabilities in the context **instructable systems**

The capabilities **real or simulated environment--artificial environment** and **other instruction--only examples** are very similar - click here if you want to enter another system to distinguish them Distinguish

The systems **etar** and **office clerk** are very similar - click here if you want to enter another capability to distinguish them Distinguish

You can elicit another capability using a triad of systems Triad

If you want specific systems included, select them in the list below

You can delete, edit, add and show matches among systems

```
etar
art
metamouse
office clerk
smallstar
noddy
tels
protos
purrpuss
```

You can delete, edit, add and show matches among capabilities

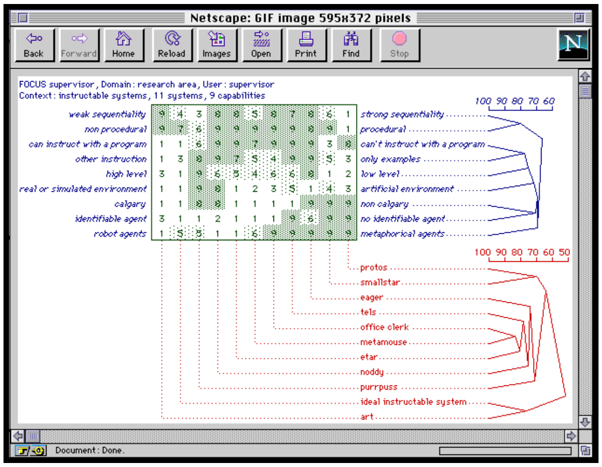
```
only examples--other instruction
low level--high level
real or simulated environment--artificial environment
procedural--non procedural
weak sequentiality--strong sequentiality
can't instruct with a program--can instruct with a program
metaphorical agents--robot agents
no identifiable agent--identifiable agent
calgary--non calgary
```

You can save or display the grid Save Display PrinCom FOCUS

You can edit the terms Edit **You can send us a comment** Comment

Figure 10 WebGrid continuing display of elements, constructs and functionality

Many other options are also offered in Figure 10. The elements and constructs are shown in sub-windows where one or more may be selected by clicking upon them, and the user may chose to delete, edit or add elements and constructs, or display the matches between them. The user may also choose to display the grid or develop a model of the relations between elements and constructs using the PrinCom or FOCUS clustering techniques. Both of these generate a colored graphical presentation of the results, and WebGrid converts this to the CompuServe GIF format and sends it back to the browser for display as shown in Figures 11 and 12.



FOCUS supervisor, Domain: research area, User: supervisor
Context: instructable systems, 11 systems, 9 capabilities

weak sequentiality	9	4	3	9	9	5	9	7	9	6	1	strong sequentiality
non procedural	9	7	6	9	9	9	9	9	9	9	1	procedural
can instruct with a program	1	1	6	9	9	7	9	9	9	3	9	can't instruct with a program
other instruction	1	3	9	9	7	5	4	9	9	5	3	only examples
high level	3	1	9	6	5	4	6	6	9	1	2	low level
real or simulated environment	1	1	9	9	1	2	3	5	1	4	3	artificial environment
calgary	1	1	9	9	1	1	1	1	9	9	9	non calgary
identifiable agent	3	1	1	2	1	1	1	9	6	9	9	no identifiable agent
robot agents	1	5	5	1	1	6	9	9	9	9	9	metaphorical agents

protos
smallstar
eager
tels
office clerk
metamouse
etar
noddy
purrpuss
ideal instructable system
art

Figure 11 WebGrid FOCUS cluster analysis

FOCUS sorts the grid for proximity between similar elements and similar constructs. From Figure 11 it can be seen that the constructs “weak sequentiality—strong sequentiality” and “non procedural—procedural” are seen as related, as are the elements “office clerk” and “metamouse.”

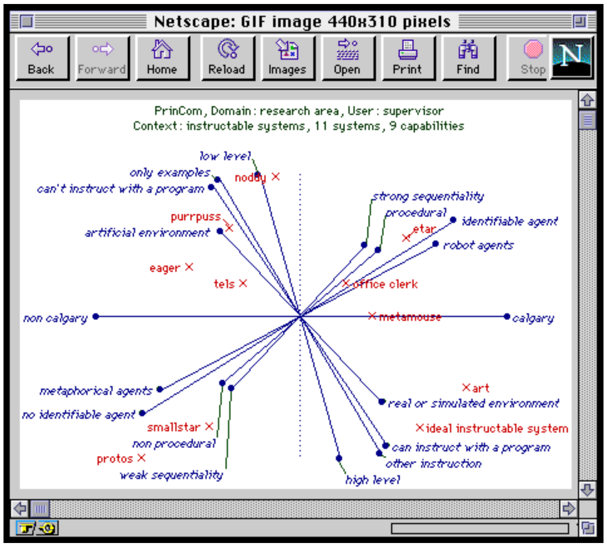


Figure 12 WebGrid PrinCom principal components analysis

PrinCom uses principal component analysis to represent the grid in minimum dimensions. From Figure 12 it can be seen that there are two clusters of related constructs typified by “low level—high level” and “procedural—non procedural” respectively, plus an isolated construct “calgary—non calgary” raising the issue that Calgary work is seen to be “procedural” and “high level.”

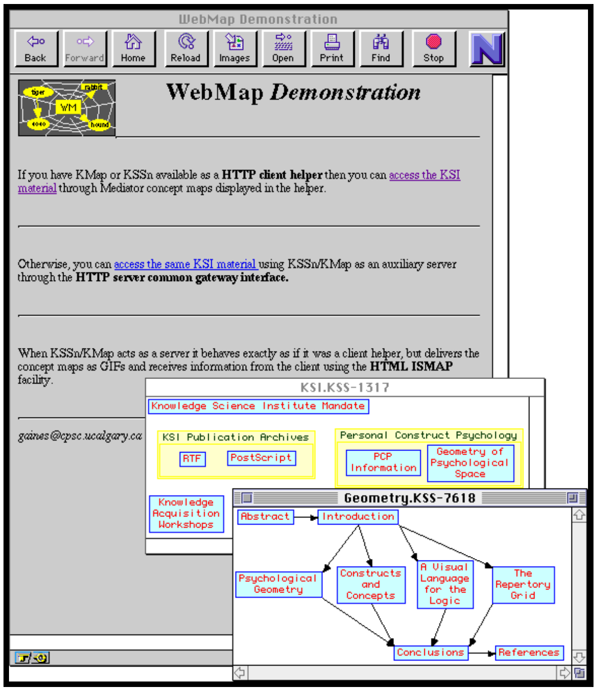
The user may also choose to save the grid locally on the client machine. Since the protocol is stateless, and all the grid data is stored in hidden input fields in the HTML form, this is simply a matter of saving the HTML source at the local machine. WebGrid includes the server url in the HTML form so that the file saved may be reloaded at any time and the interaction continued without the need to take special action at either client or server.

5 Conceptual Mapping on the Web

Concept maps are used to structure argument forms and express relationships between ideas (Gaines and Shaw, 1993b). In education, Nowak and Gowin (1984) have promoted the use of concept maps to investigate a student’s understanding of a topic, and there are many different forms that have been applied in this field (Lambiotte, Dansereau, Cross and Reynolds, 1989). In management, Axelrod (1976) proposed *cognitive 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 maps that came to be termed *semantic networks* and used extensively for formal knowledge representation.

We have developed a general concept mapping tool called KMap (Gaines and Shaw, 1993b, 1994) which provides a grapher for nodes and arcs that can be programmed by the user to support different forms of concept map. User interaction with KMap takes place through the creation of statements in the visual language, and through interaction with such statements through popup menus whose content is specific to node type. The action initiated is context-sensitive: to the node selected for the popup, to nodes linked to it, and to other nodes preselected by clicking on them. This allows complex activities to be initiated by natural user actions.

WebMap is a development of KMap to operate on the web as both a client helper and an auxiliary server. As a client helper, KMap is 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 web browser. Figure 13 shows an HTML document in Netscape which has links to KMap documents.



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/KMap acts as a server it behaves exactly as if it was a client helper, but delivers the concept maps as GIFs and receives information from the client using the **HTML ISMAP** facility.

gaines@cpsc.ucalgary.ca

Figure 13 KMap acts a client helper application to the NetScape browser

When the user clicks on the hypertext link “access the KSI material” at the top of the screen, Netscape fetches a KMap file “KSI.KSS” which it passes to KMap which open and displays it as shown in the KMap window at the middle right.

Clicking on the node “Geometry of Psychological Space” at the right of this map causes KMap to send a message to Netscape requesting that the file with url “http://ksi.cpsc.ucalgary.ca/WebMap/Geometry.KSS” be fetched. This is a concept map of the structure of an article that opens in another KMap window shown at bottom right. Clicking on the node “Constructs and Concepts” in the center of this map causes KMap to send a message to Netscape requesting that the url “http://ksi.cpsc.ucalgary.ca/PCP/PCPIntro.html#4” be fetched. This is a section of an HTML document that NetScape then displays. Clicking on other nodes in the concept map causes NetScape to navigate to different sections of this document.

KMap is currently implemented only for the Apple Macintosh and hence can act as a client helper only on Macintosh computers. We are currently working on ports to Windows and Motif 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 noneditable concept maps as *clickable maps* in HTML documents. We have interfaced KMap as an auxiliary server through the *common gateway interface* to the WebStar server to allow the same concept maps to be used as clickable maps.

Figure 14 shows the file fetched when the user clicks on the “access the same KSI material” in the second paragraph of the document of Figure 13. KMap converts the concept map to a GIF file and delivers it as an image that Netscape can read.

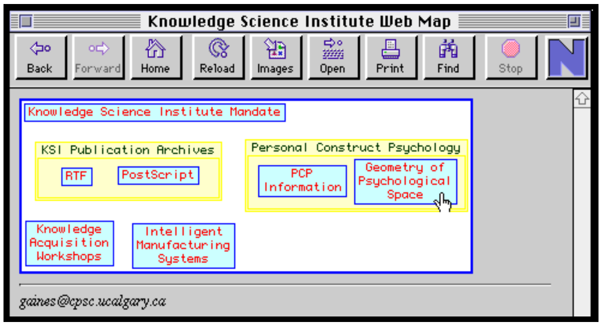

Knowledge Science Institute Web Map
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
Knowledge Science Institute Mandate
KSI Publication Archives
RTF
PostScript
Personal Construct Psychology
PCP Information
Geometry of Psychological Space
Knowledge Acquisition Workshops
Intelligent Manufacturing Systems
gaines@cpsc.ucalgary.ca

Figure 14 KMap acting as a server through the common gateway interface to WebStar

Clicking on the node “Geometry of Psychological Space” sends the coordinates of the point clicked back to KMap on the server which then takes the same action as if the map had been clicked in a helper. Thus KMap supports the use of concept maps on World Wide Web through client helpers and server gateways in an integrated way.

6 Conclusions

The implementation of a Mediator system to support a learning web has been described. It uses list servers to support the discourse processes among members of the community, web servers to support the dissemination of knowledge through documents and multimedia resources, and repertory grid and conceptual map auxiliary web servers to support the elicitation, sharing and comparison of conceptual structures.

The primary capability missing in Mediator currently is that of exploring the consequences of knowledge through simulation. Recent enhancements to the web through languages such as JAVA (Sun, 1995) offer programmable components that can be sent from a server to a client and be executed at the client. This provides the capability to support dynamic simulations, and we are exploring the potential of such new facilities to extend the implementation of the learning web.

Acknowledgements

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada.

References

Andreessen, M. (1993). NCSA Mosaic Technical Summary. NCSA, University of Illinois. ftp://ncsa.uiuc.edu/Web/Mosaic/Papers/mosaic.ps.Z.

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

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

Berners-Lee, T. (1993). World-Wide Web Talk at Online Publishing 1993. CERN, Geneva. http://info.cern.ch/hypertext/WWW/Talks/OnlinePublishing93/Overview.html.

Berners-Lee, T., Connolly, D. and Muldrow, K. (1994). Hypertext Markup Language (HTML) Version 2.0. CERN, Geneva. http://www.hal.com/products/software/olias/Build-html/html-spec.ps.

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

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., Norrie, D.H. and Lapsley, A.Z. (1995). Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise. **Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics**. pp.964-969. New York, IEEE.

Gaines, B.R. and Shaw, M.L.G. (1989). Comparing the conceptual systems of experts. **Proceedings of the Eleventh International Joint Conference on Artificial Intelligence**. pp.633-638. San Mateo, California, Morgan Kaufmann.

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). 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). Concept maps indexing multimedia knowledge bases. **AAAI-94 Workshop: Indexing and Reuse in Multimedia Systems**. pp.36-45. Menlo Park, California, AAAI.

Gaines, B.R. and Shaw, M.L.G. (1995a). Collaboration through concept maps. Schnase, J.L. and Cunnius, E.L., Ed. **Proceedings of CSCL95: Computer Support for Collaborative Learning**. pp.135-138. Mahwah, New Jersey, Lawrence Erlbaum.

Gaines, B.R. and Shaw, M.L.G. (1995b). Concept maps as hypermedia components. **International Journal Human-Computer Studies 43**(3) 323-361.

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

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

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

Maturana, F.P. and Norrie, D.H. (1995). A generic mediator for multi-agent coordination in a distributed manufacturing system. **Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics**. pp.952-957. New York, IEEE.

Norrie, D.H. and Gaines, B.R. (1995). The learning web: An agent-oriented network supporting knowledge access,collaboration and simulation. Maurer, H., Ed. **Proceedings of ED-MEDIA'95 : World Conference on Educational Multimedia and Hypermedia**. pp.500-505. Charlottesville, VA, Association for the Advancement of Computing in Education.

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

Popper, K.R. (1968). Epistemology without a knowing subject. Rootselaar, B. Van, Ed. **Logic, Methodology and Philosophy of Science III**. pp.333-373. Amsterdam, North-Holland.

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

Raggett, D. (1994). A review of the HTML+ document format. **Proceedings of WWW94**. pp.http://www1.cern.ch/PapersWWW94/dsr.ps.

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. and Gaines, B.R. (1995). Comparing constructions through the web. Schnase, J.L. and Cunnius, E.L., Ed. **Proceedings of CSCL95: Computer Support for Collaborative Learning**. pp.300-307. Mahwah, New Jersey, Lawrence Erlbaum.

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