Participatory Design and Validation of Knowledge-Based Systems through the Web

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

Abstract

This article describes experience in porting tools for eliciting, modeling and validating knowledge structures to operate through the World Wide Web. The objective is to support knowledge-based system development and application with distributed communities of experts and users. It is concluded that it would be valuable for those within the knowledge acquisition, knowledge modeling, validation and verification, and knowledge-based systems application communities to collaborate to provide a rich web of facilities for the development and operation of knowledge-based systems.

1 Introduction

One lesson learnt in the development of knowledge based systems has been the importance of acquiring and managing knowledge from multiple sources of expertise (Boose and Bradshaw, 1987; Gaines and Shaw, 1989; Shaw and Gaines, 1989). In particular, the use of multiple sources makes possible some degree of relatively independent cross-validation of the knowledge models developed. It also highlights anomalies where experts disagree in the interpretation of 'well-established' terminology and concepts.

In software engineering in general it has also been found important to involve a wide spectrum of users in the early system design. Methodologies such as participatory design (Schuler and Namioka, 1993) and joint application design (August, 1991) emphasize the need to use structured system development methodologies that give end-users central roles in the design process. It is common to find that user terminology and concepts do not correspond to those of domain experts, analysts and designers.

As the techniques of knowledge acquisition, modeling and validation become increasingly widely applied to a wide range of system developments, it is becoming important to make the methodologies and tools available on a community basis. In particular, new users should be able to calibrate themselves against existing practice in the use of terminology and concepts critical to the cognitive interface between the system and its users.

The World Wide Web (Berbers-Lee, Calico, Luotonen, Nielsen and Secret, 1994) provides a widely accessible technology for client-server computing supporting geographically distributed communities of users. The hypermedia capabilities of the web provide support for managing the diversity of background knowledge involved in knowledge-based system development. The interactive graphical user interfaces capabilities of the web offer the capability to support knowledge acquisition, modeling, validation and application tools.

This article describes experience in porting tools for eliciting, modeling and validating knowledge structures to operate through the World Wide Web. The objective is to support knowledge-based system development and application involving distributed communities of experts and users.

2 The Architecture of the World Wide Web

The World Wide Web was conceived by Berners-Lee in March 1989 as a "hypertext project" to organize documents at CERN (Berners-Lee and Cailliau, 1990). The design involved: a simple hypertext markup language that authors could enter through a word processor; distributed servers running on machines anywhere on the network; and access through any terminal.

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 radically changed the nature of the web. Andreessen (1993) issued NCSA Mosaic version 2 using tags to encode definitions of Motif widgets embedded within a hypermedia document, and allowed the state of those widgets within the client to be transmitted to the server. With the addition of such *forms* providing graphic user interface (GUI) capabilities, the web protocols became the basis of general interactive, distributed, client-server information systems.

Knowledge acquisition systems can be implemented on the web as: *client helpers* running locally on the same machine as the web browser and communicating with it through its application program interface (API); or *auxiliary servers* running remotely on the same machine or network as the web server and communicating with it through its common gateway interface (CGI).

The advantages of client helper implementation are that existing knowledge acquisition tools can be used, and the full power of the local machine GUI is available. The disadvantages are that the tool needs to be implemented for all platforms, minimally UNIX, PC and Mac, and the tool has to be distributed to all users.

The advantages of auxiliary server implementation are that a single implementation for one server platform only needs to be developed and maintained, and the program does not have to be issued to other sites. The disadvantages are the limited capabilities of the GUIs encoded in HTML forms and the transmission delays of the network.

Figure 1 gives an overview of what is involved in factoring an application to operate in client-server mode through the web using existing web browsers as clients.

At the left the client component consists of a web browser providing a graphic-user interface through:

- *HTML documents* supporting typographic text
- *Images* supporting graphics, possibly embedded in the text
- *Hyperlinks* attached to text or images such as icons
- *Clickable maps* supporting returning to the server the location of a mouse click in an image
- *HTML forms* providing a programmable GUI returning data entered to the server.

At the left center, the client communicates through the net with a web server that is either an implementation of the HTTP protocol in the application server component, or a standard HTTP server providing a gateway to the application server component.

At the right the server component splits into 4 parts:

- Encoding the application output as HTML and decoding the incoming POST parameters
- Representation of the application output as a document and extracting incoming data
- Data processing as the non-interface parts of the application
- Parameters as separate data structures or scripts providing application variants and dialog in different languages.

In addition, at the lower left the persistent storage for the application has to be factored to be represented either in hidden fields which can be stored as an HTML document at the client or in files or a database stored at the server, or some combination of both.

The remainder of the article illustrates the port of a knowledge acquisition tool used for previous studies of cross-validation of terminology and concepts in a community of experts (Shaw and Gaines, 1989) with examples drawn from data from those studies.

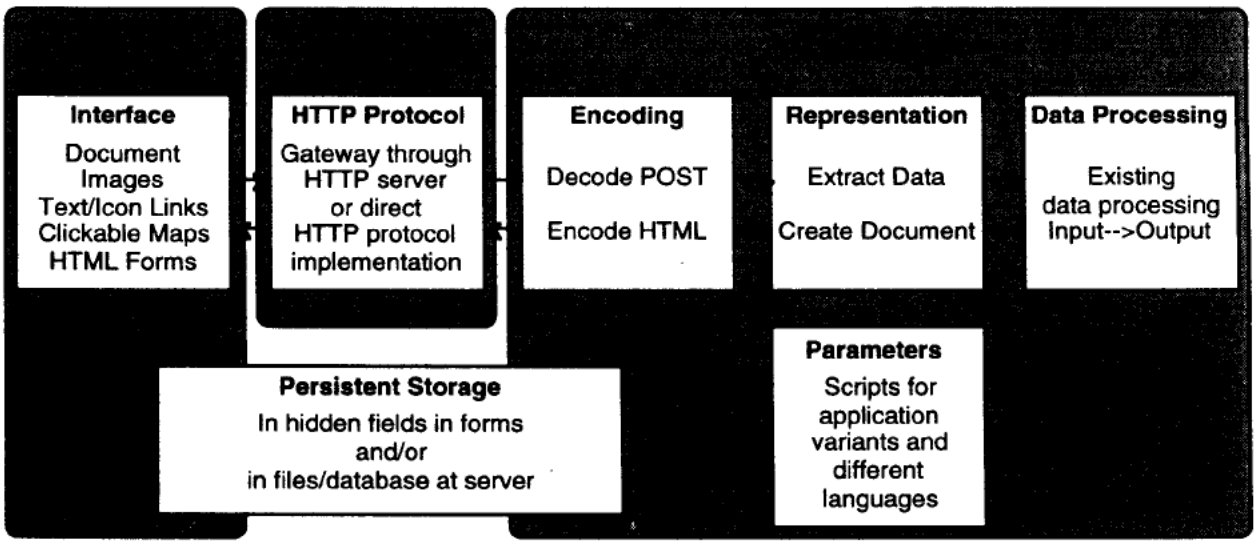

Figure 1 Factoring an application for client-server operation through the web

3 WebGrid: Porting a Repertory Grid Tool to the Web

Repertory grids based on personal construct psychology (Kelly, 1955) have been used for knowledge acquisition since the early years of knowledge-based system development (Shaw and Gaines, 1983; Boose, 1984) and have been refined over the years to support increasingly complex knowledge structures (Boose, Bradshaw, Koszarek and Shema, 1993; Ford, Bradshaw, Adams-Webber and Agnew, 1993; Gaines and Shaw, 1993).

Since grid elicitation tools can be used directly by experts and end-users it would be very useful to have them accessible through any personal computer or workstation with access to the web. This is feasible because the primary data input format is through rating scales, and this can be done effectively by using popup menus which are available through the HTML GUI.

Figure 2 shows the initial screen of a repertory grid elicitation tool operating through the web using a standard web browser and requiring no client helpers.

The HTML form requests the usual data required to initiate grid elicitation: user name; domain and context; terms for elements and constructs; default rating scale; and a list of initial elements.

When the user clicks on the "Done" button, the browser transmits the data entered to the remote server which passes it through its CGI to a specialist knowledge acquisition auxiliary server. The server processes the data and generates an HTML document that it returns to the browser resulting in the screen shown in Figure 3 eliciting a construct from a triad of elements.

The user clicks on a radio button to select an element which she construes as different from the other two, enters terms characterizing the construct, and clicks on "Done". The server generates the screen shown on the left of Figure 4 which places a popup menu rating scale alongside each element enabling the user to rate each one along the new construct as shown on the right. She is also able to change the terms used if they seem inappropriate in the context of all the elements, and to change the ratings of the already entered elements if appropriate.

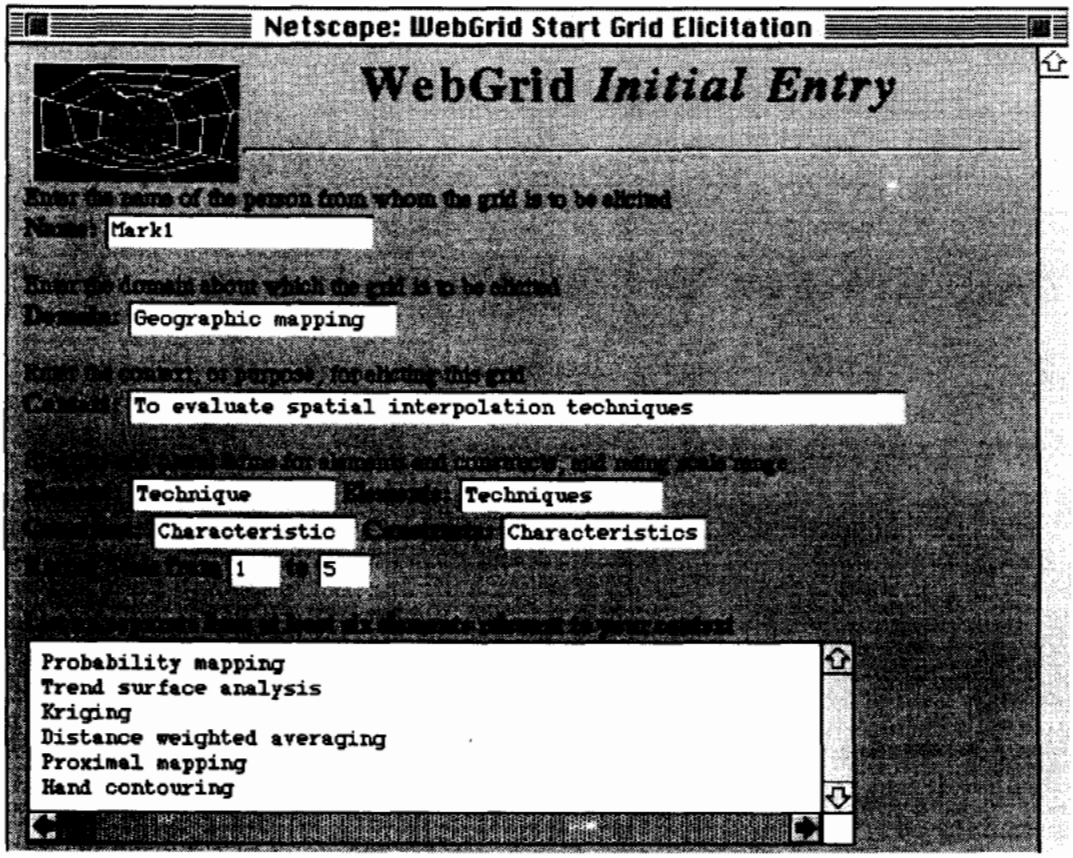

Figure 2 Starting an elicitation

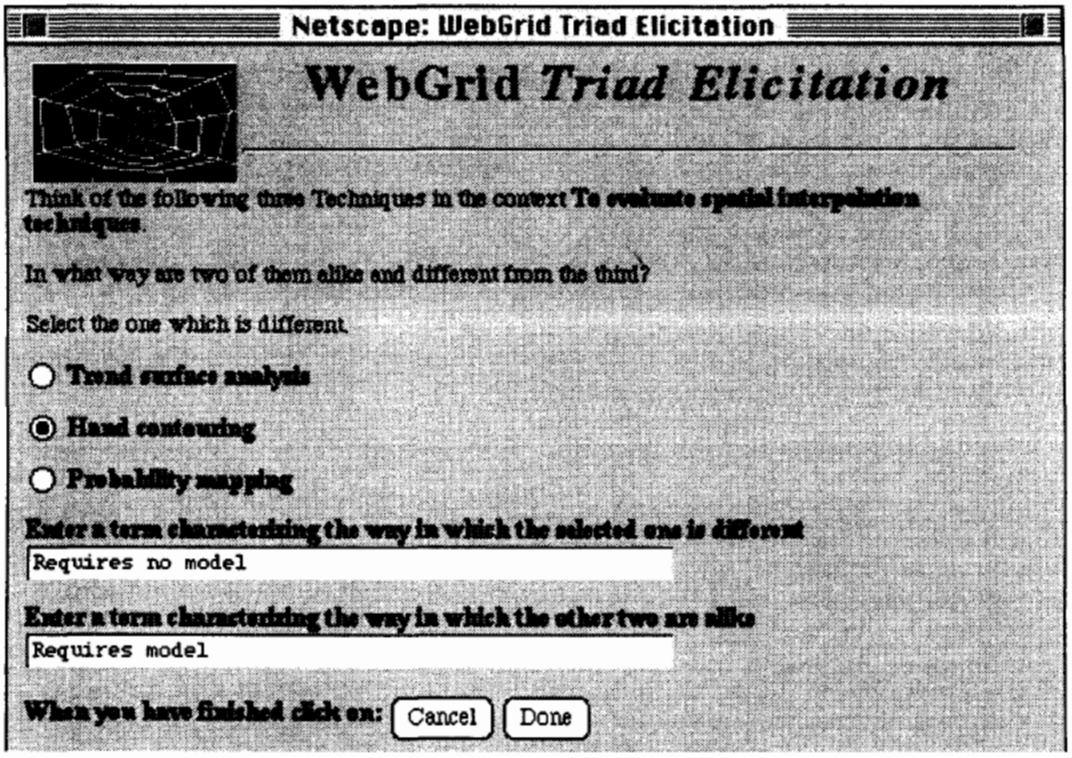

Figure 3 Initial triadic elicitation of a construct

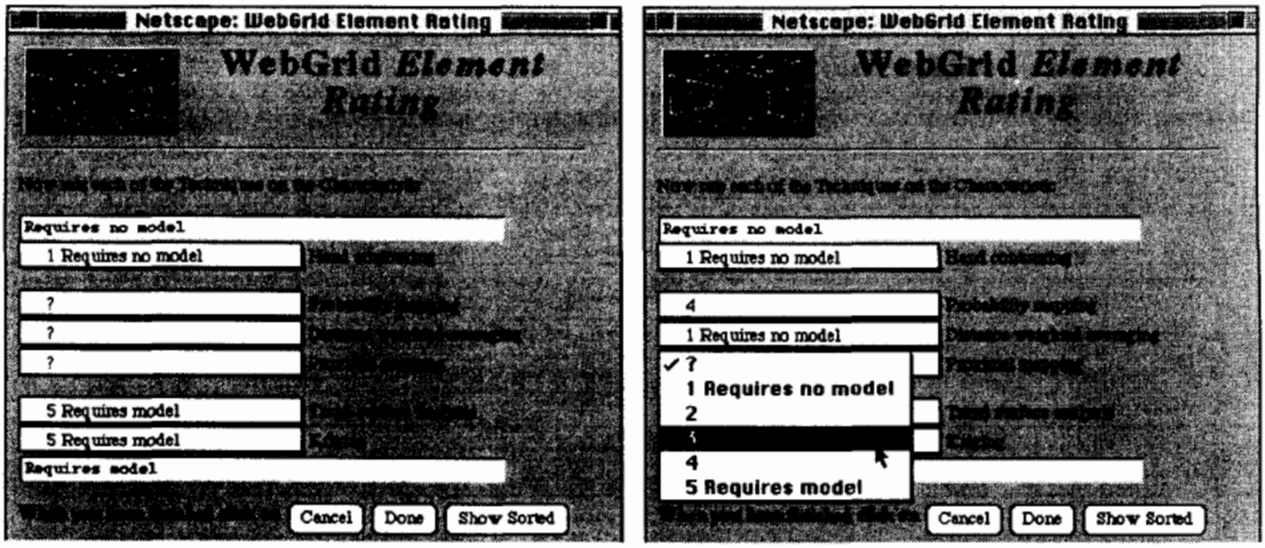

Figure 4 Rating other elements after triadic elicitation using popup menu as a rating scale

Clicking on the "Done" button in Figure 4 sends the ratings back to the server which generates a status screen listing elements and constructs and offering various ways of continuing the elicitation or analyzing the data. Figure 5 shows this status screen after the elicitation of 11 elements and 12 constructs.

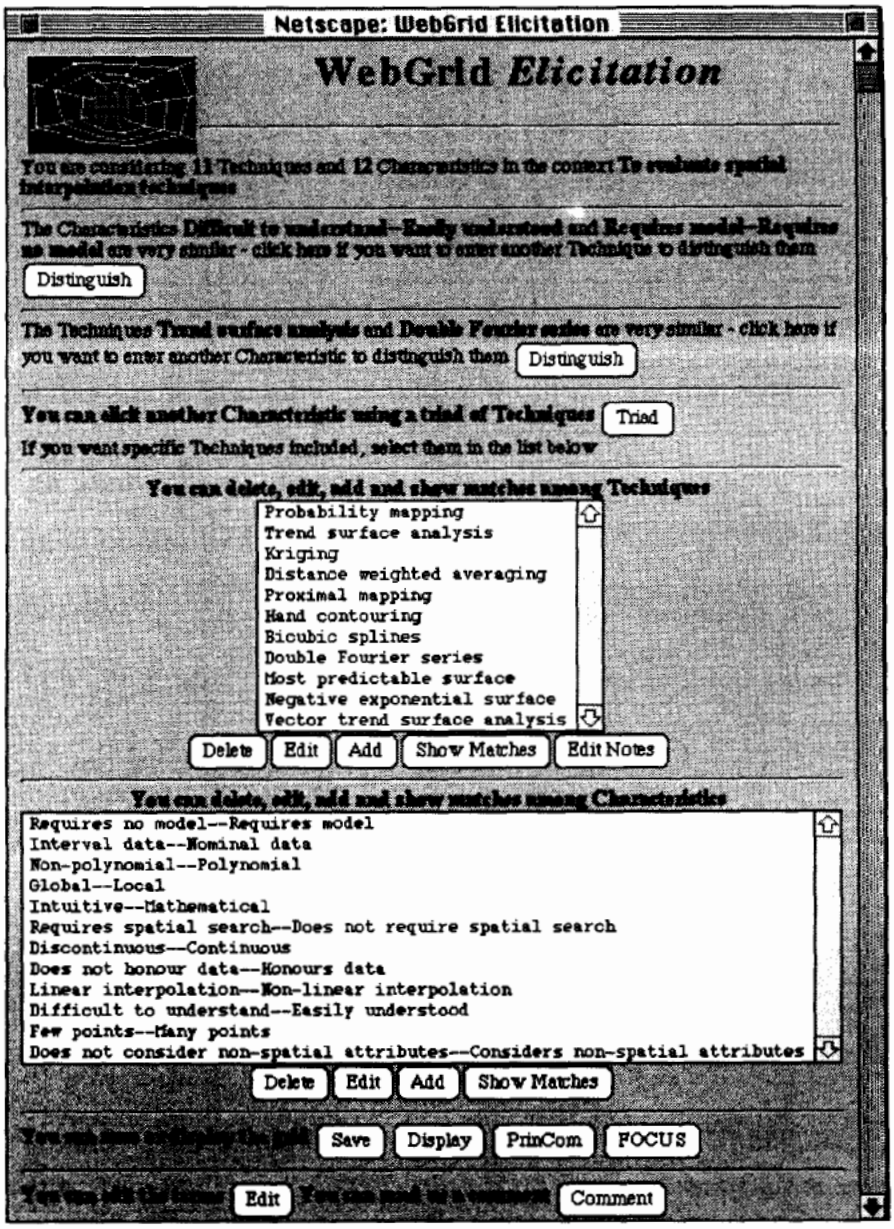

Figure 5 Status screen after elicitation of 11 elements and 12 constructs

The various sections of the status screen are generated dynamically from the knowledge that has been elicited. They draw the user's attention to: matching constructs, suggesting a new element be added to break the match; matching elements suggesting a new construct be added to break the match; and the opportunity to elicit another construct from a triad of elements. The elements and constructs are listed in selection boxes allowing selected items to be deleted and edited, new items to be added, matches to be shown, and HTML annotation to be added to elements.

Further buttons enable the data elicited to be saved or to be displayed or analyzed in various ways. The results of analysis are usually presented as graphic output. This is generated at the server, converted to GIF format and returned to the client where it can be examined and saved if required.

igure 6 shows the output returned when the "FOCUS" button is used to develop a hierarchical knowledge model clustering related constructs and elements (Shaw, 1980).

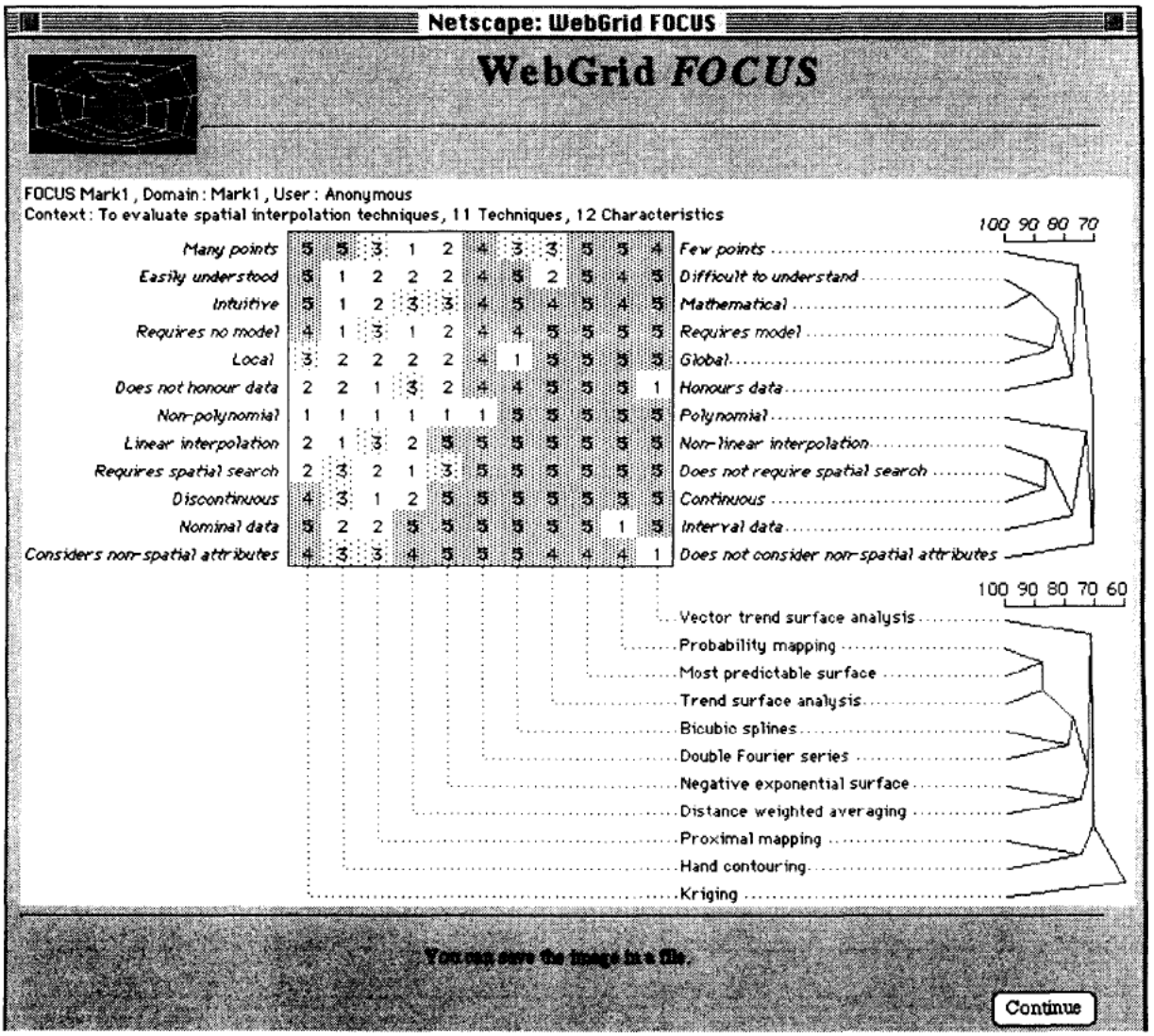

Figure 6 FOCUS cluster analysis of grid elicited

4 Cross Validation of Elicited Knowledge Structures

Once knowledge acquisition tools are available on the web it becomes possible to support distributed acquisition in which users at different sites can compare their terminology and concepts. WebGrid supports the comparison of knowledge structures using an extension of the consensus-conflict-correspondence-contrast methodology (Shaw and Gaines, 1989). When a grid has been developed at one site, another person at another site can develop a grid based on it, either attempting to rate the elements on the other expert's constructs to evaluate consensus and conflict, or developing her own constructs to evaluate correspondence and contrast.

Figure 7 shows the initial screen for an expert entering ratings in a grid using the elements and constructs from another's grid.

When the expert has entered their name and clicked on "Done" the server generates screens for rating similar to Figure 4, and elicitation proceeds as before.

A "Compare" button in the status screen returns the "exchange" analysis shown in Figure 8.

The graph of declining matches on the right shows that there is consensus on the construct "interval data—nominal data" but conflict on "linear interpolation—non-linear interpolation."

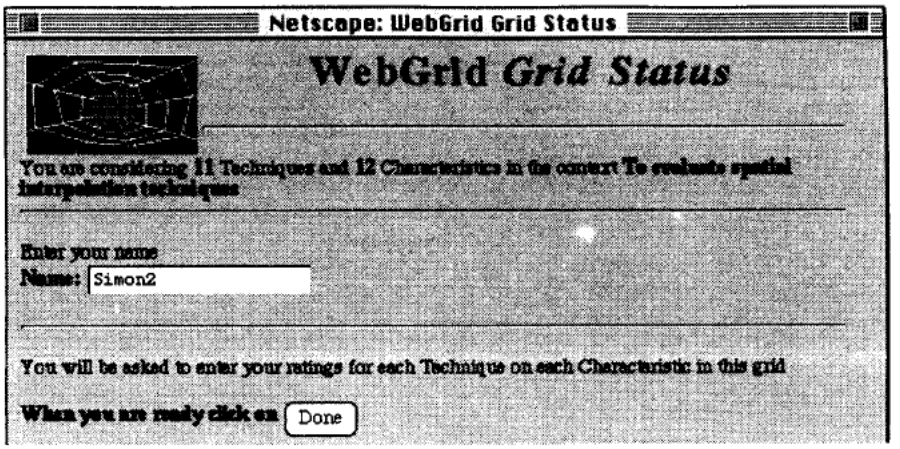

Figure 7 Commencing an exchange grid

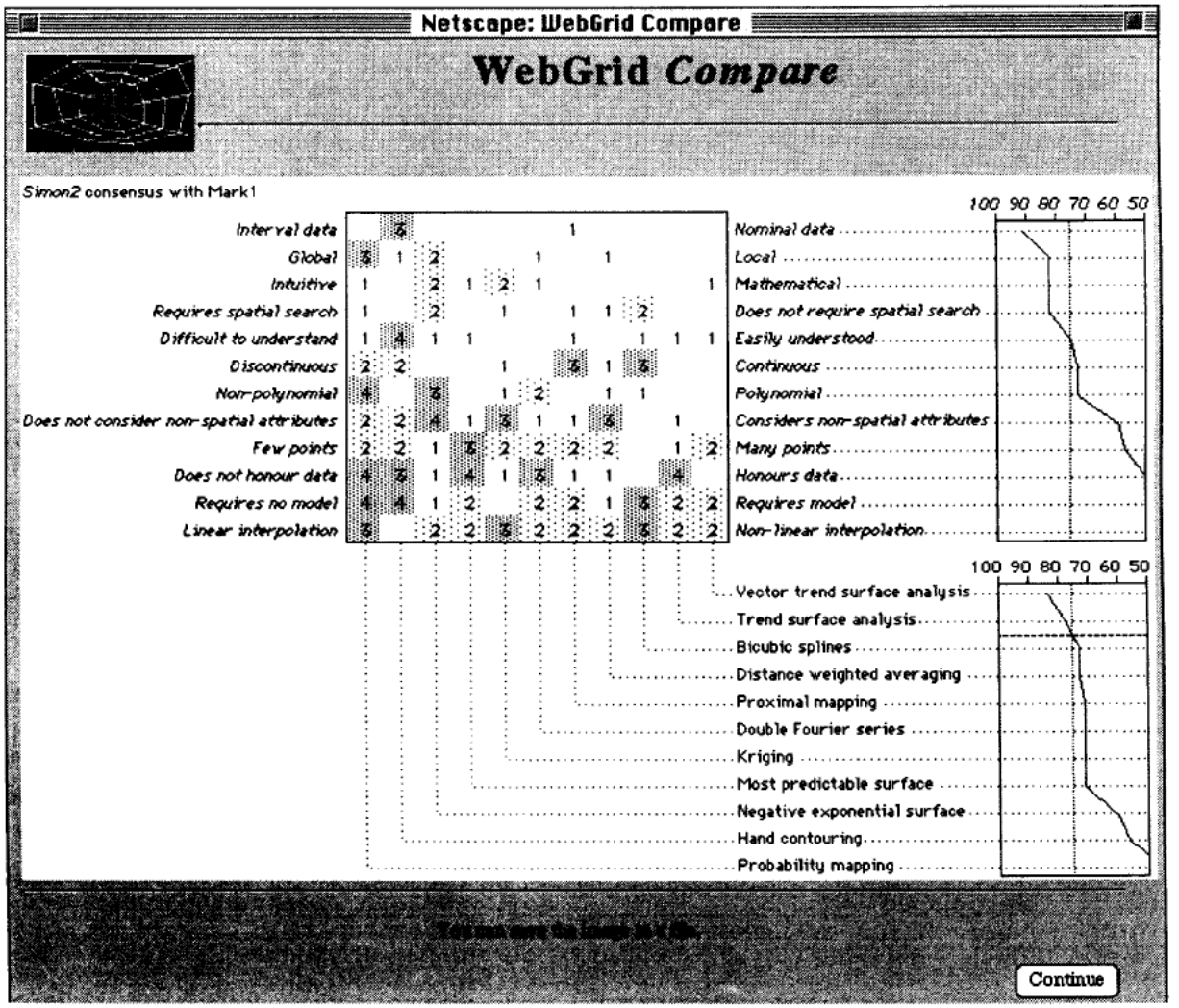

Figure 8 Exchange analysis of conflict and consensus

WebGrid also allows one expert to commence with the elements from another expert's grid and develop constructs through triads, breaking matches, and so on.

Figure 9 shows the analysis of the correspondence between two grids relating to the same element. Each construct in one grid is linked to the most closely corresponding construct in the other grid using the pattern of ratings, not the names.

The graph of declining matches on the right shows that there is correspondence between the constructs "local—global" in each grid, and, more interestingly, between "interval or ratio data—nominal data" and "high level data—low level data".

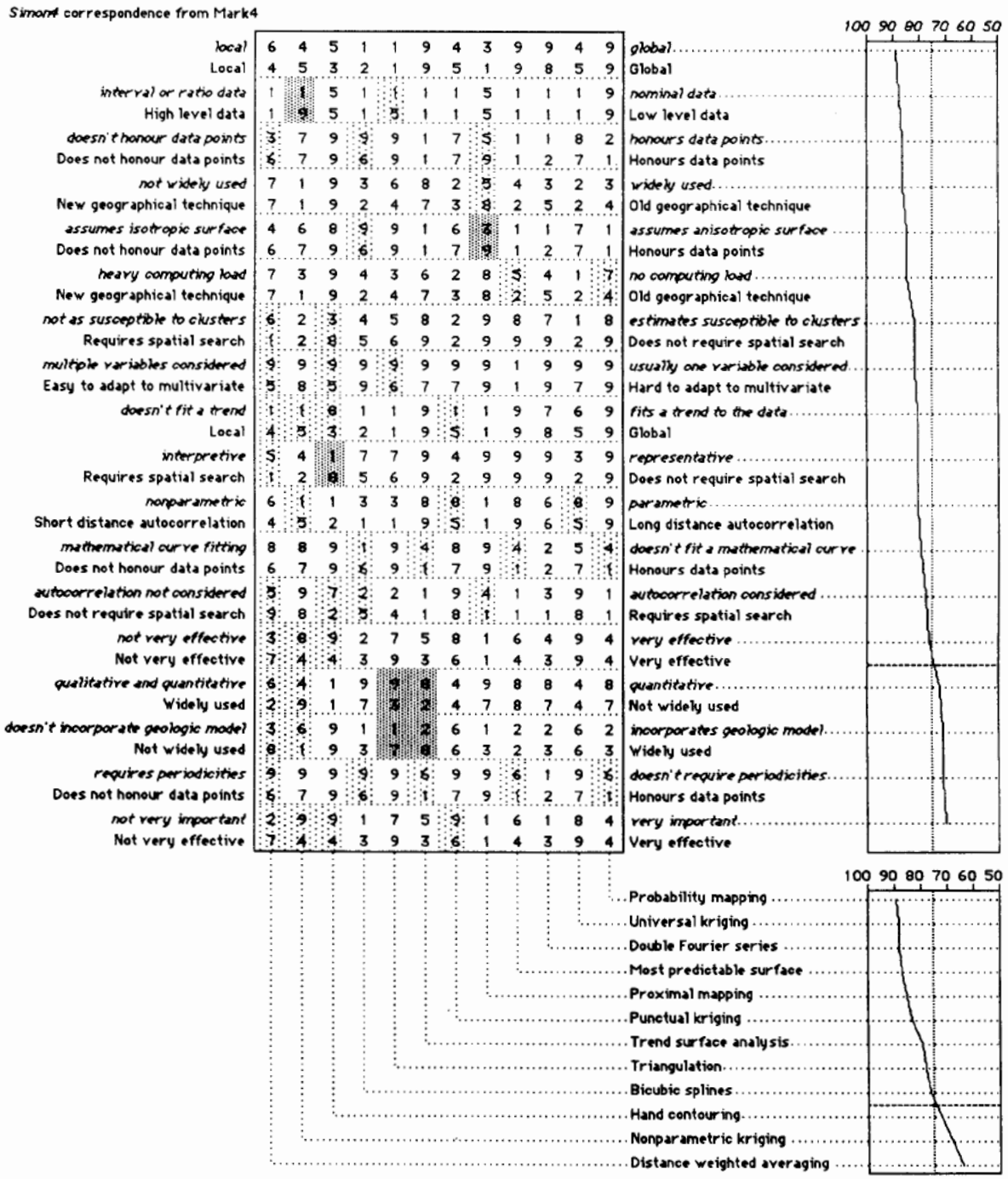

Figure 9 Analysis of correspondence and contrast

The elicitation techniques and analyses of Figures 8 and 9 implement the four way comparison of constructs and terminology shown in Figure 10 (Gaines and Shaw, 1989; Shaw and Gaines, 1989). This provides for cross-validation of the constructs and terminology being used in a knowledge-based system, both between experts and between experts and clients. The web implementation makes this validation methodology readily available in a distributed community, and the graphic displays of the analyses make them simple to understand by those developing and using the knowledge-based system.

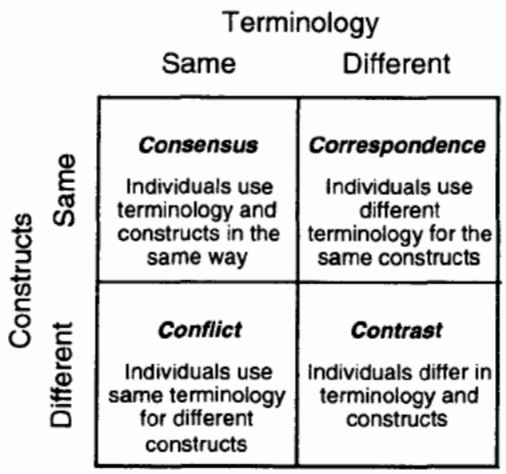

Figure 10 Four-way comparison of knowledge structures in terms of the constructs and the terminology used for them

5 Related Research

A number of knowledge-based systems have been developed on the web. The SHADE (SHAred Dependency Engineering) project (McGuire, Kuokka, Weber, Tenenbaum, Gruber and Olsen, 1993) supports reconfigurable exchange of engineering knowledge among computer-aided engineering systems by using agents providing services through the Internet. These agents are generally accessible through the KQML agent-communication language (Finin, Weber, Wiederhold, Genesereth, Fritzson, McKay, McGuire, Shapiro and Beck, 1992) and have been used to support systems operating through the web. For example, Gruber and Gauthier (1993) have used them in providing a Device Modeling Environment (DME) on the web, and have demonstrated its use in the simulation of a leak in the Space Shuttle's reaction control system using clickable maps as mimic diagrams. Rice, Farquhar, Piernot and Gruber (1995, 1996) have developed a web interface to the Ontolingua ontology construction tool, and Farquhar, Fikes, Pratt and Rice (1995) have given examples of its application to collaborative ontology construction for information integration. Eriksson (1995) has made some of the PROTÉGÉ-II knowledge-acquisition tools accessible through the web.

6 Conclusions

The graphic user interface and client server capabilities of the World Wide Web can support the use of knowledge acquisition, modeling and validation methodologies and tools in a distributed user community. Such tools can be made available as widely accessible services over the Internet, and tools from different developers may be used together to provide new forms of integrated facilities.

A move to the client-server environment of the web is not just a change in the technology of knowledge-based system implementation. It provides the basis for the development of a next generation of knowledge-based systems in which acquisition and validation are not just phases in system development but are instead fully integrated into ongoing system application.

The problem with systems developed to operate on a stand-alone basis is that the knowledge engineering activity becomes separated from the system application. The client-server model enables the clients to be widely distributed, emulating the stand-alone model, but the server for a particular application to be centralized supporting ongoing system monitoring and improvement. Thus, experience gained in the continuing application of the system at many sites can be consolidated through an ongoing knowledge acquisition and validation activity that is supported as part of the system design and operation.

It has been proposed that validation be incorporated in knowledge-based system operation through anomaly detection rules that trigger reports when a rule or constraint is violated (Gaines, 1988). The difficulty with this approach in stand-alone systems is that there is no natural channel of communication for such reports. In the client-server model, the server remains under the control of the knowledge engineering team and anomalies are reported and the relevant context captured at the server.

There are obvious potential problems of security and privacy with a client-server model, as there are with any distributed networked system. However, the problems are ones commonly faced by those operating corporate data bases of any kind, and security procedures and supporting technologies are available to manage them.

In conclusion, it would be valuable for those within the knowledge acquisition, knowledge modeling, validation and verification, and knowledge-based systems application communities to collaborate to provide a rich web of facilities for the development and operation of knowledge-based systems.

Acknowledgments

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

URL for WebGrid

WebGrid can be accessed online at
http://tiger.cpsc.ucalgary.ca/WebGrid

References

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

August, J.H. (1991). **Joint Application Design: The Group Session Approach to System Design**. EngleWood Cliffs, New Jersey, Yourdon Press.

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. and Cailliau, R. (1990). WorldWideWeb: Proposal for a Hypertext Project. CERN, Geneva. http://info.cern.ch/hypertext/WWW/Proposal.html.

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

Boose, J.H. (1984). Personal construct theory and the transfer of human expertise. **Proceedings AAAI-84**. pp.27-33. California, American Association for Artificial Intelligence.

Boose, J.H. and Bradshaw, J.M. (1987). Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for knowledge-based systems. **International Journal of Man-Machine Studies 26** 3-28.

Boose, J. H., Bradshaw, J. M., Koszarek, J.L. and Shema, D. B. (1993). Knowledge acquisition techniques for group decisions support. **Knowledge Acquisition 5**(4) 405-447.

Eriksson, H. (1995). PROTÉGÉ-II Knowledge-Acquisition Tools. http://camis.stanford.edu/projects/protege/ka-tools.html.

Farquhar, A., Fikes, R., Pratt, W. and Rice, J. (1995). Collaborative ontology construction for information integration. Knowledge Systems Laboratory, Stanford University. ftp://www-ksl-svc.stanford.edu:5915/.

Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McKay, D., McGuire, J., Shapiro, S. and Beck, C. (1992). Specification of the KQML Agent-Communication Language. The DARPA Knowledge Sharing Initiative External Interfaces Working Group.

Ford, K.M., Bradshaw, J.M., Adams-Webber, J.R. and Agnew, N.M. (1993). Knowledge acquisition as a constructive modeling activity. **International Journal of Intelligent Systems 8**(1) 9-32.

Gaines, B.R. (1988). Positive feedback processes underlying the formation of expertise. **IEEE Transactions on Systems, Man & Cybernetics SMC-18**(6) 1016-1020.

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.

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

Gruber, T. and Gautier, P. (1993). Machine-generated explanations of engineering models: a compositional modeling approach. **IJCAI'93: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence**. pp.1502-1508 (http://www-dme.stanford.edu:5051/DME-server/DME-server/Computed/TOP-LEVEL).

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

McGuire, J.G., Kuokka, D.R., Weber, J.C., Tenenbaum, J.M., Gruber, T.R. and Olsen, G.R. (1993). SHADE: Technology for knowledge-based collaborative engineering. **Journal of Concurrent Engineering: Applications and Research 1**(2)

Rice, J., Farquhar, A., Piernot, P. and Gruber, T. (1995). Stanford KSL Network Services. Knowledge Systems Laboratory, Stanford University. http://www-ksl-svc.stanford.edu:5915/.

Rice, J., Farquhar, A., Piernot, P. and Gruber, T. (1996). Lessons learned using the web as an application interface. **Proceedings of CHI'96**. (ftp://ksl.stanford.edu/pub/KSL_Reports/KSL-95-69.ps). New York, ACM.

Schuler, D. and Namioka, A., Ed. (1993). **Participatory Design**. Hillsdale, New Jersey, Erlbaum.

Shaw, M.L.G. (1980). **On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models Of The World**. London, Academic Press.

Shaw, M.L.G. and Gaines, B.R. (1983). A computer aid to knowledge engineering. **Proceedings of British Computer Society Conference on Expert Systems**. pp.263-271. Cambridge, British Computer Society.

Shaw, M.L.G. and Gaines, B.R. (1989). A methodology for recognizing conflict, correspondence, consensus and contrast in a knowledge acquisition system. **Knowledge Acquisition 1**(4) 341-363.