Comparing Constructions through the Web

Mildred L G Shaw and Brian R Gaines

Knowledge Science Institute, University of Calgary

Abstract

Much of the richness of a collaborative learning environment comes from the differing constructions that different learners bring to the learning domain. However, the differences in terminology and its usage that stem from different construct systems can also be major impediments to collaboration. Techniques from personal construct psychology may be used to make such tacit differences overt and a source of rich discussion among collaborative learners. This article describes the use of construct elicitation, modeling and comparison services on the World-Wide Web to enable collaborative learners to understand one another's constructs.

Keywords — Personal construct psychology, repertory grids, World-Wide Web, collaborative learning.

1. Introduction

Personal construct psychology (PCP) (Kelly, 1955) emphasizes the idiographic nature of individual constructions of the world and hence raises questions about the basis of inter-personal collaboration and shared, social constructions (Shaw, 1985). PCP has been used extensively in both studying and supporting learning processes (Shaw, 1979; Pope and Shaw, 1981; Pope and Keen, 1981; Shaw and Gaines, 1992), and computer-based tools have been developed to elicit, model and compare construction systems (Shaw, 1980; Shaw and Gaines, 1989; Gaines and Shaw, 1993). Such tools running on personal computers have been used to support collaborative learning (Shaw and Gaines, 1992), and the tools have been extended to support collaborative elicitation and analysis over local area networks (Shaw and Gaines, 1991; Shaw and Gaines, 1993).

Tools for eliciting and comparing construction systems could become a truly *emancipatory technology* (Habermas, 1968) for collaborative learning, if they were widely available over the Internet. This article describes the operation of such tools as part of the World-Wide Web, and their application to collaborative learning.

2. Construct Elicitation and Modeling

The major methodology that we have used for the elicitation of constructs and terminology from individuals and groups is based on extensions of the *repertory grid* technique originally proposed by Kelly (1955) as an empirical measurement methodology appropriate to personal construct psychology. Repertory grid techniques elicit knowledge indirectly by prompting individuals for critical elements and relevant constructs in a coherent sub-domain. The techniques are difficult to undertake manually as they require feedback and management from the elicitor while at the same time attempting to avoid inter-personal interactions that would distort the elicitee's construct structures. Hence the advent of the personal computer in the mid-1970s and its evolution into the graphic workstations of the 1980s has made the computer implementation of interactive repertory grid elicitation an attractive area of development (Shaw, 1980; Shaw, 1981; Mancuso and Shaw, 1988).

The repertory grid methodology gives a basis for approximating intensional distinctions, or *constructs*, through their extensions when applied to elements in a domain. The distinctions made by two individuals can then be compared in terms of the differences in their extensions and in the terminology used. The two relations of similarity between distinctions and between terminology give rise to a 4-way classification of constructs (Gaines and Shaw, 1989; Shaw and Gaines, 1989) as shown in Figure 1.

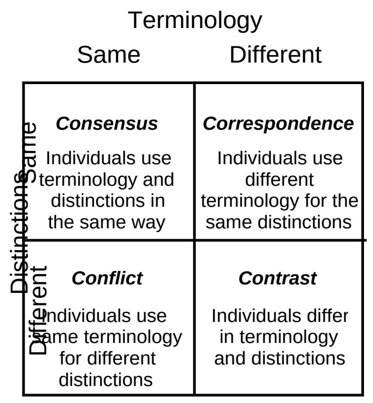

Terminology
Same
Different
Distinctions
Same
Different
Consensus
Individuals use terminology and distinctions in the same way
Correspondence
Individuals use different terminology for the same distinctions
Conflict
Individuals use same terminology for different distinctions
Contrast
Individuals differ in terminology and distinctions

Figure 1 Four-way comparison of constructs in terms of the distinctions made and the terminology used for them

Consensus arises if the construct systems assign the same term to the same distinction. *Conflict* arises if the systems assign the same term to different distinctions. *Correspondence* arises if the conceptual assign different terms to the same distinction. *Contrast* arises if the systems assign different terms to different distinctions. The usual technique for eliciting these comparisons is to have two people negotiate a common set of elements characterizing a domain, each separately develop their personal constructs based on these elements, and then exchange their grids with the ratings removed and attempt to rate the elements on the other’s constructs. The comparison of the exchanged grids allows consensus and conflict be modeled, while that of the original grids allows correspondence and contrast to be modeled.

3. Computer Supported Modeling of Construction Systems

RepGrid (CPCS, 1993) is a computer-based tool providing an integrated set of tools for elicitation and analysis of elements and constructs in a given domain. It combines a number of different techniques, for construct elicitation, modeling, and comparison. The main tools in RepGrid are:

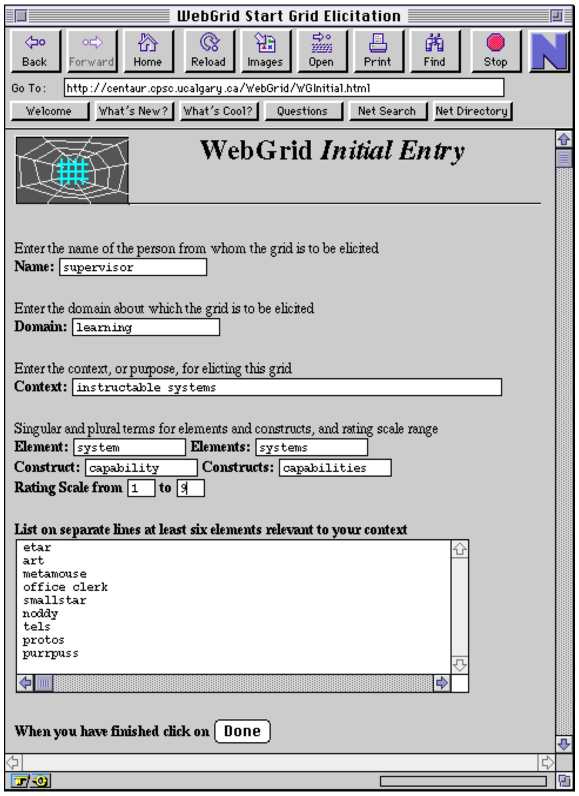
- *Elicitation tools* which accept specifications of elements within a domain and provide an interactive graphical elicitation environment within which a person can distinguish elements to derive his or her constructs within the domain. The resultant construct system is continuously analyzed to provide feedback prompting the person to enter further elements and constructs.
- *Modeling tools* use various forms of clustering and entailment derivation for the analysis and display of the construct systems elicited: FOCUS shows the system as a hierarchical structure; and PrinCom as a spatial map.
- *Exchange tools* extend the elicitation tools to share elements and constructs between people and allow the terms in the construct system derived from one person to be used by another in order to determine whether the two construct systems are different in any way. It can also be used by the same person looking at changes in their own construct structures over time, for example: after reading a specific book, or exploring a particular domain.
- *Comparison tools* process results from several people to reveal the similarities and differences in their construct systems, or the same person at different times, construing a domain defined through common elements or constructs. It can be used to focus discussion among people on the differences which require resolution, enabling them to classify the disparities in terms of differing terminologies, levels of abstraction, disagreements, misunderstandings, and so on.

WebGrid is a port of RepGrid to operate as a service over World-Wide Web (WWW). Part of its functionality is an interface to RepGrid through the HTTP common gateway interface to MacHTTP. Other significant parts are a dialog generator that replaces the graphic user interface to RepGrid on personal computers with dynamically generated HTML forms accessible through WWW browsers, and a high-speed PICT-to-GIF converter that supports the transmission across the web of the graphic output from RepGrid analyses.

WebGrid is being used to support collaborative learning activities in undergraduate and graduate courses at the University of Calgary. The following sections illustrate its application and show the way in which the personal construct systems of learners may be elicited, modeled and compared through computer-based tools.

4. WebGrid in Action

Figure 2 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.



http://centaur.cpsc.ucalgary.ca/WebGrid/WGInitial.html

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

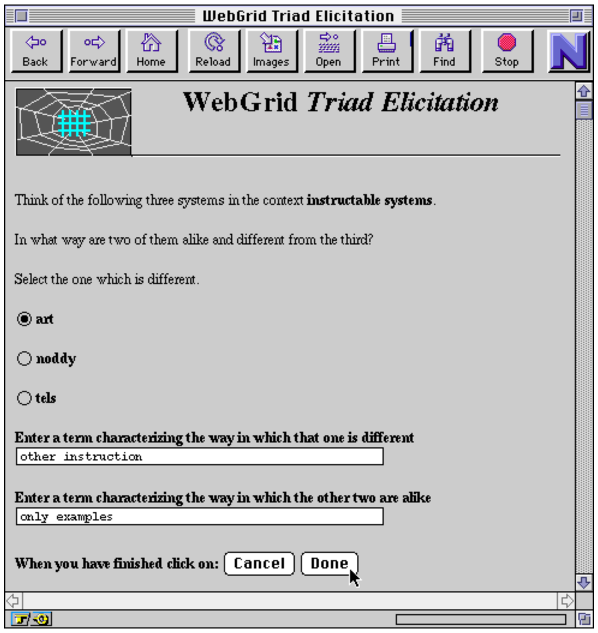
List on separate lines at least six elements relevant to your context

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

When you have finished click on Done

Figure 2 WebGrid initial entry screen

When the supervisor has entered the data shown in Figure 2, he clicks on “Done” and WebGrid generates the screen in Figure 3 where he is asked to distinguish three of his elements using the standard RepGrid 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

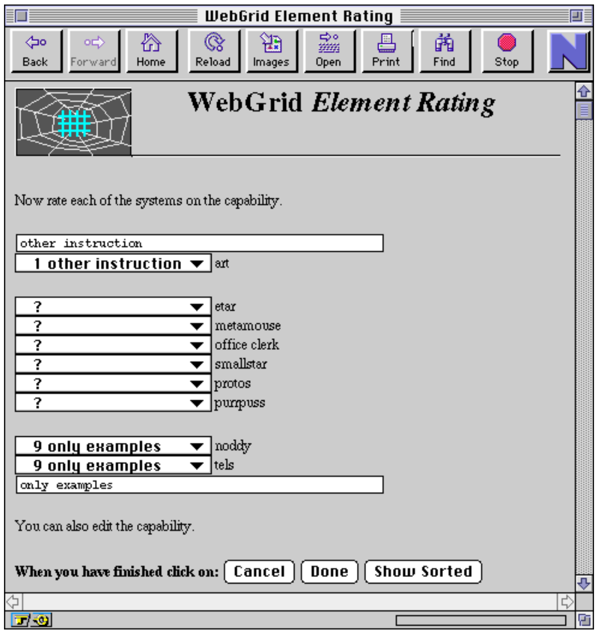
Enter a term characterizing the way in which that one is different
other instruction

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

When you have finished click on: Cancel Done

Figure 3 WebGrid elicitation of a construct from a triad of elements

He clicks on "Done" and WebGrid generates HTML for the screen in Figure 4 where the elements are shown alongside popup menus which can be used to rate them on the new construct as shown in Figure 5.



WebGrid *Element Rating*

Now rate each of the systems on the capability.

other instruction

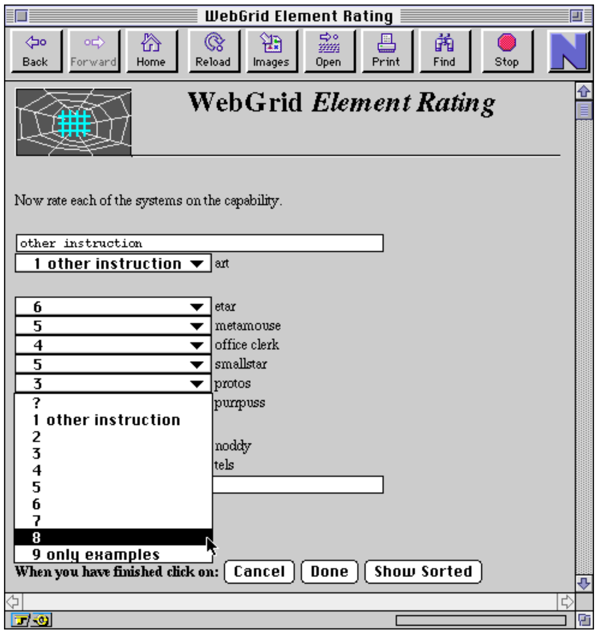
1 other instruction	art
?	etar
?	metamouse
?	office clerk
?	smallstar
?	protos
?	purrpuss
9 only examples	noddy
9 only examples	tels

only examples

You can also edit the capability.

When you have finished click on: Cancel Done Show Sorted

Figure 4 WebGrid rating of elements on a new construct



WebGrid *Element Rating*

Now rate each of the systems on the capability.

other instruction

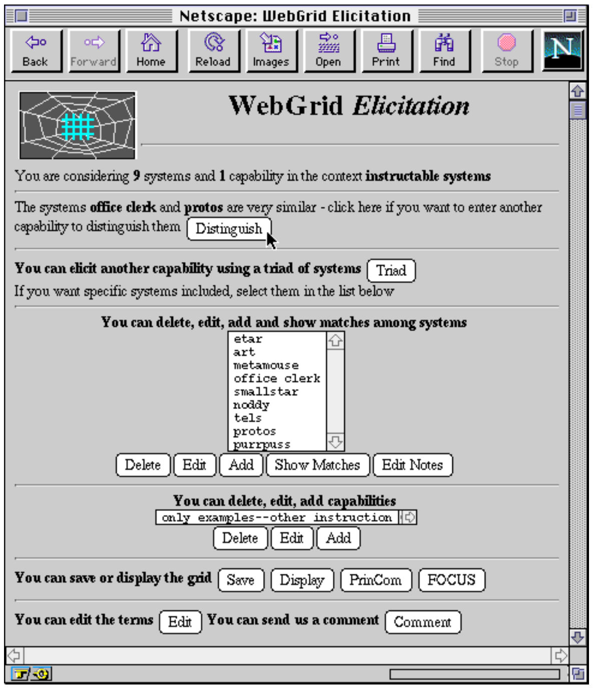
1 other instruction	art
6	etar
5	metamouse
4	office clerk
5	smallstar
3	protos
	purrpuss
	noddy
	tels

? 1 other instruction 2 3 4 5 6 7 8 9 only examples

When you have finished click on: Cancel Done Show Sorted

Figure 5 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 6 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

Delete Edit Add Show Matches Edit Notes

You can delete, edit, add capabilities
only examples--other instruction
Delete Edit Add

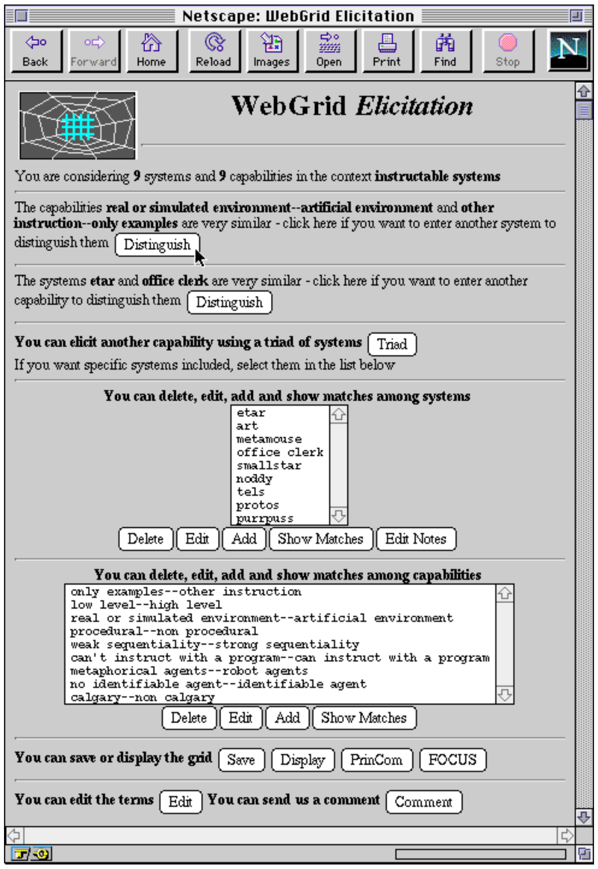
You can save or display the grid Save Display PrinCom FOCUS

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

Figure 6 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 4.

Figure 7 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.”



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
```

[Delete] [Edit] [Add] [Show Matches] [Edit Notes]

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
```

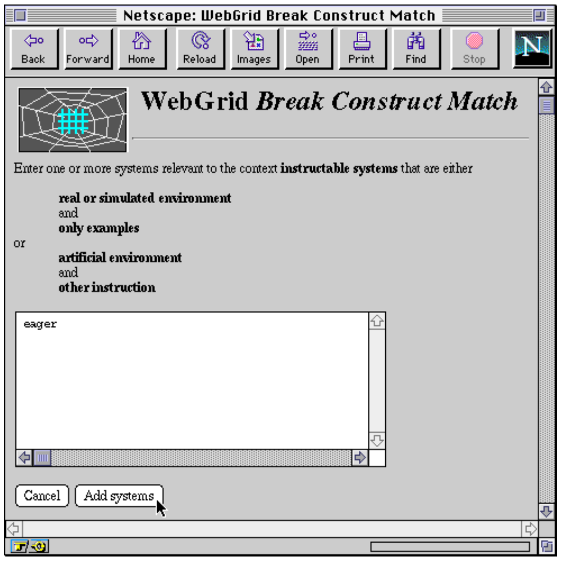
[Delete] [Edit] [Add] [Show Matches]

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 7 WebGrid continuing display of elements, constructs and functionality

If the user clicks on “Distinguish” WebGrid generates HTML for the screen shown in Figure 8 where the user is explicitly asked to add one or more new elements that are either “weak sequentiality” and “procedural”, or “strong sequentiality” and “non procedural.” Thus WebGrid guides the elicitation through feedback about relevant actions that the user may take, but the system is strongly non-modal in that no particular action is forced upon the user.


Netscape: WebGrid Break Construct Match
Back
Forward
Home
Reload
Images
Open
Print
Find
Stop
WebGrid Break Construct Match

Enter one or more systems relevant to the context **instructable systems** that are either

real or simulated environment
and
only examples

or

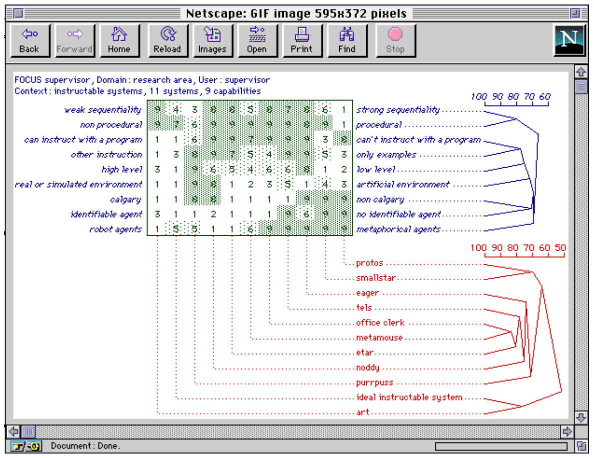
artificial environment
and
other instruction

```
eager
```

[Cancel] [Add systems]

Figure 8 WebGrid entering a new element to break a construct match

Many other options are also offered in Figure 7. 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 9 and 10.



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

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

Figure 9 WebGrid FOCUS cluster analysis

FOCUS sorts the grid for proximity between similar elements and similar constructs. From Figure 9 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.”

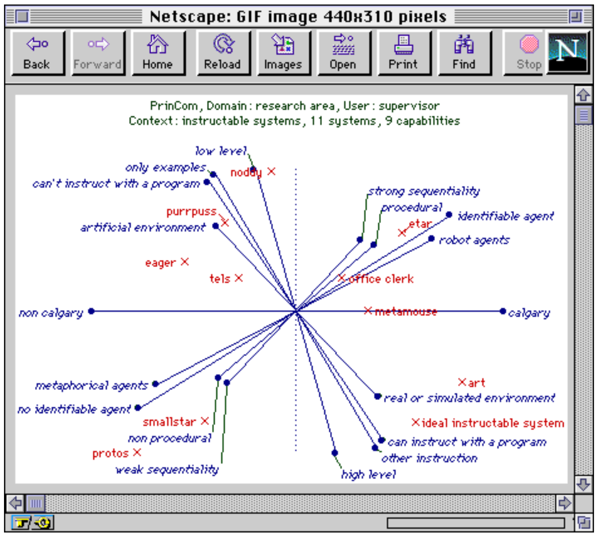


Figure 10 WebGrid PrinCom principal components analysis

PrinCom uses principal component analysis to represent the grid in minimum dimensions. From Figure 10 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. WebGrid in Collaborative Learning

The examples given have illustrated WebGrid in use by an individual developing a personal model of a domain. This is a useful exercise in individual learning that readily extends to groups since RepGrid has techniques for the comparison of construct systems enabling collaborative learners to explore similarities and differences in the way they construe a domain.

WebGrid allows a user to commence an elicitation based on another person's grid, either using just the elements in it and developing his or her own constructs, or using both elements and constructs but commencing with all the rating unknown (an "exchange" grid). In the first case the new grid may be compared with the original one to determine what constructs correspond in the two grids—the users may be making the same distinctions but giving them different names. In the second case the new grid may be compared with the original one to determine to what extent the ratings correspond in the two grids—the users may be making different distinctions but giving them the same names. When an elicitation is based on an existing grid an additional analysis button appears in the screens of Figures 6 and 7, "Compare."

The example given so far is taken from a graduate class which encouraged students to clarify their interactions with their supervisors, committees and other students in related areas. A graduate student of the supervisor who developed the grid above used the elements in this grid to develop his own constructs. Figure 11 shows the graph returned by WebGrid when the student clicks on "Compare" after having added one construct.

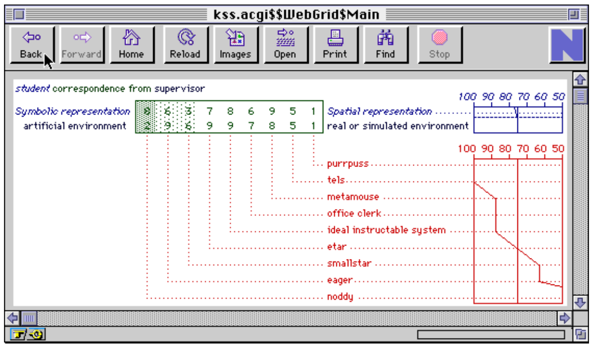
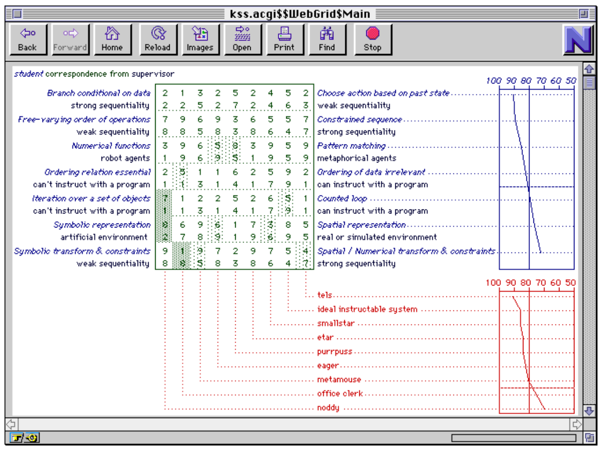


Figure 11 WebGrid correspondence in supervisor's constructs to student's first construct

The RepGrid analysis selects the supervisor's construct that most closely matches that of the student and displays it below that of the student, adding graphs of the matches between constructs and elements. The student can see that when he makes the distinction "symbolic representation—spatial representation" the supervisor has available a closely related distinction that he terms "artificial environment—real or simulated environment." This raises questions about the nature of the match: is it just differing terminology; is it differing levels of abstraction; is it a causal link; is it a correlated effect in the data?

Figure 12 shows the WebGrid comparison of the student's completed grid with that of his supervisor. For each construct in the student's grid the analysis has selected that in the supervisor's grid that makes a similar distinction among the elements, regardless of what that construct is called. Thus the student's construct "branch conditional on data—choose action based on past state" is shown as corresponding to the supervisor's construct "strong sequentiality—weak sequentiality." In this case the student's construct is concrete and operational whereas the supervisor's is abstract, and they may both gain by understanding and discussing the difference in terminology and the basis for it. An analysis such as that of Figure 12 may often lead to prolonged discussion over a period and to new research insights or the resolution of disagreement—the foundations of collaborative learning.



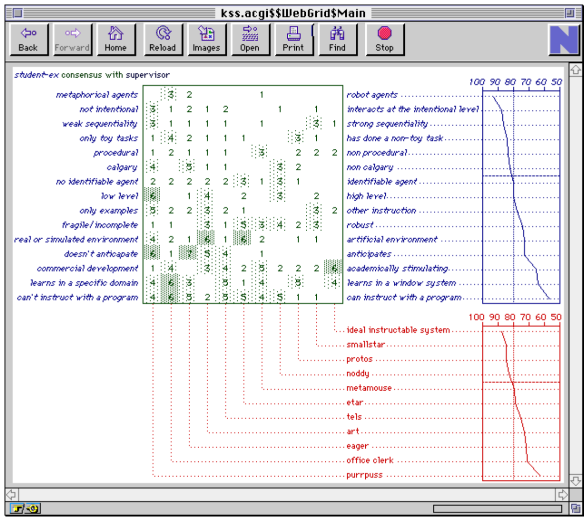
kss.acgi$$WebGrid$Main

student correspondence from supervisor

	noddy	office clerk	metamouse	eager	purrpuss	etar	smallstar	ideal instructable system	tels	
Branch conditional on data	2	1	3	2	5	2	4	5	2	*Choose action based on past state*
strong sequentiality	2	2	5	2	7	2	4	6	3	weak sequentiality
Free-varying order of operations	7	9	6	9	3	6	5	5	7	*Constrained sequence*
weak sequentiality	8	8	5	8	3	8	6	4	7	strong sequentiality
Numerical functions	3	9	6	5	8	3	9	5	9	*Pattern matching*
robot agents	1	9	6	9	5	1	9	5	9	metaphorical agents
Ordering relation essential	2	5	1	1	6	2	5	9	2	*Ordering of data irrelevant*
can't instruct with a program	1	1	3	1	4	1	7	9	1	can instruct with a program
Iteration over a set of objects	7	1	2	2	5	2	6	5	1	*Counted loop*
can't instruct with a program	1	1	3	1	4	1	7	9	1	can instruct with a program
Symbolic representation	8	6	9	6	1	7	3	8	5	*Spatial representation*
artificial environment	2	7	8	9	1	9	6	9	5	real or simulated environment
Symbolic transform & constraints	9	1	9	7	2	9	7	5	4	*Spatial / Numerical transform & constraints*
weak sequentiality	8	8	5	8	3	8	6	4	7	strong sequentiality

Figure 12 WebGrid correspondence in supervisor's constructs to all student's constructs

After having developed his own grid, the student also used WebGrid in exchange mode to rate all the elements on his supervisor's constructs. Figure 13 shows the comparison of the student's ratings of the elements on the constructs entered by his supervisor. This analysis allows them to determine to what extent they agree in the use of the same terminology in their research. It is apparent, for example, that they do not agree on the use of the construct "can't instruct with a program—can instruct with a program" or on the characterization of the instructable system "purrpuss."



kss.acgi$$WebGrid$Main

student-ex consensus with supervisor

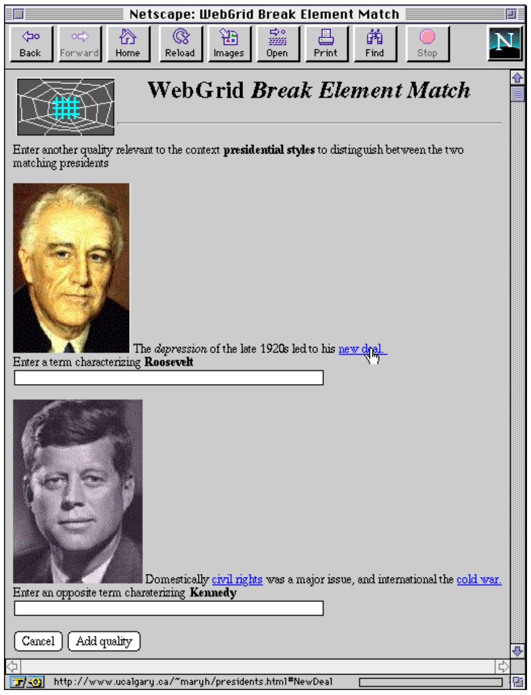
	purrpuss	office clerk	eager	art	tels	etar	metamouse	noddy	protos	smallstar	ideal instructable system	
metaphorical agents		3	2				1					*robot agents*
not intentional	3	1	2	1	2			1		1		*interacts at the intentional level*
weak sequentiality	3	1	1	1	1		1			3	1	*strong sequentiality*
only toy tasks	1	4	2	1	1	1			3	1		*has done a non-toy task*
procedural	1	2	1	1	1		3		2	2	2	*non procedural*
calgary	4		5	1	1			3	2			*non calgary*
no identifiable agent	2	2	2	2	2	3	1	3	1			*identifiable agent*
low level	6		1	4		2		3		2		*high level*
only examples	5	2	2	3	2	1				3	2	*other instruction*
fragile/incomplete	1	1		3	1	5	3	4	2	3		*robust*
real or simulated environment	4	2	1	6	1	6	2		1	1		*artificial environment*
doesn't anticapate	6	1	7	5	4		1					*anticipates*
commercial development	1	4		3	4	2	5	2	2	2	6	*academically stimulating*
learns in a specific domain	4	6	3		5	1	4		5		4	*learns in a window system*
can't instruct with a program	4	6	5	2	5	5	4	5	1	1		*can instruct with a program*

Figure 13 WebGrid consensus between student and supervisor ratings

Note that the supervisor and student undertook all these activities reciprocally, each developing their own grids for the domain independently using their choice of elements, and the supervisor also exchanged with the student and rated all the student's elements on the student's constructs. This is important, because one objective in collaborative learning is to reduce the impact of the power differential between "teacher" and "learner" and create a single "learning community."

6. Linking to MultiMedia on the Web

A system operating through WWW should be capable of using the hypermedia facilities of the web. WebGrid allows annotation to be added to any element ("Edit Notes" button in Fig. 7). This annotation is in HTML so that it can include diagrams, pictures and links to other web documents. Figure 14 shows a grid being developed on presidential policies where the elements "Eisenhower" and "Kennedy" match, and the student has decided to add another construct to break the match. The HTML annotation includes pictures of each president and hypertext links to major issues during their terms of office. Figure 15 shows the annotations that generate the images and links shown in Figure 14.



Netscape: WebGrid Break Element Match

WebGrid *Break Element Match*

Enter another quality relevant to the context **presidential styles** to distinguish between the two matching presidents

The *depression* of the late 1920s led to his new deal.

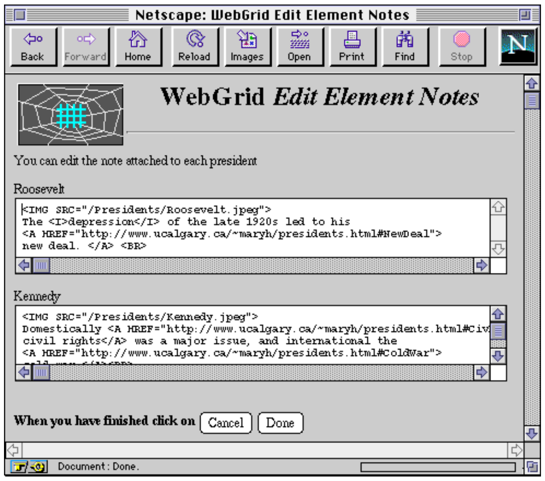
Enter a term characterizing **Roosevelt**

Domestically civil rights was a major issue, and international the cold war.

Enter an opposite term charaterizing **Kennedy**

http://www.ucalgary.ca/~maryh/presidents.html#NewDeal

Figure 14 WebGrid links to multimedia annotation



Netscape: WebGrid Edit Element Notes

WebGrid *Edit Element Notes*

You can edit the note attached to each president

Roosevelt

```
<IMG SRC="/Presidents/Roosevelt.jpeg">
The <I>depression</I> of the late 1920s led to his
<A HREF="http://www.ucalgary.ca/~maryh/presidents.html#NewDeal">
new deal. </A> <BR>
```

Kennedy

```
<IMG SRC="/Presidents/Kennedy.jpeg">
Domestically <A HREF="http://www.ucalgary.ca/~maryh/presidents.html#Civ
civil rights</A> was a major issue, and international the
<A HREF="http://www.ucalgary.ca/~maryh/presidents.html#ColdWar">
```

When you have finished click on

Document: Done.

Figure 15 WebGrid annotation entries

7. Experience with WebGrid in Courses

Constructivist tools that allow students to investigate the relativity and pluralism of perspectives in human society are particularly important in disciplines that still teach primarily from a positivist stance (Shaw, 1994). The 'received wisdom' stance of much science teaching is inappropriate to the employment environments of computer scientists. They need to become *reflective practitioners* in Schön's (Schön, 1983) terminology, because they will need to continually adapt throughout their careers to social change in which their discipline plays a major role. Computer science teaching generally follows the tradition of "privileged knowledge" in which it is the business of the instructor to impart knowledge. Students are fed "knowledge" in measured portions, expected to digest it, and give evidence through assignments that they have done so.

A more operational view of knowledge takes a constructivist approach to learning in which understanding is based on active participation in the subject matter and reflection on conversations with others on the topics of a course. This is the Schön's "reflection-in-action" in which learning is a joint enterprise, leading to a less authoritarian model.

An example of a course in which WebGrid plays a significant role is CPSC 547, advanced information systems for Computer Science majors. Students work collaboratively in groups of 2 to 4, and make their materials available to the class of some 50 students through the web and through demonstrations. The students in this course are preparing for a new industrial infrastructure which has seen the end of the large-scale information systems divisions that used to dominate computer science employment opportunities. Hundreds of information systems professionals have already had to come to terms with a new industrial environment that emphasizes small, adaptable, entrepreneurial organizations. Our graduating students need skills that go beyond mere technical proficiency to cope with the new challenges and opportunities.

A classroom environment suitable for reflective learning is best based on the recommendations and beliefs of Carl Rogers (Rogers, 1961) for generating a positive atmosphere in which students exhibit mature everyday behavior, are less defensive, more adaptive, and more able to meet situations creatively. This involves treating each student as an individual, being available to discuss problems individually and help with students' decision-making, creating a supportive and empathic class atmosphere in which each student is given positive encouragement to discuss issues of concern, and making the instructor's own thoughts and views genuinely available for discussion.

After a few lectures, the students in CPSC 547 take over the course and run it through their own group research, presentations and demonstrations addressing major issues in advanced information systems. The students work extremely hard, are incredibly motivated and enthusiastic, achieve a very high standard of work, and think deeply not only about the technology but also about the social and ethical implications of its applications. WebGrid allows them to explore different perspectives on the world in a structured fashion leading to results that can be presented to others and discussed within the groups and with the class as a whole.

8. Conclusions and Future Directions

When we commenced the development of WebGrid, knowing the limitations of HTML forms, we had no great expectations that the resultant tool would be attractive to use. We have spent many years carefully developing user interfaces with good human factors targeted on repertory grid elicitation, and knew the need for special-purpose graphical "widgets" that could not be emulated in HTML. We are enthusiastic users of World-Wide Web for publication, and use it extensively in courses where each instructor and student has a home page and together develop a "learning web." We were interested to see what could be achieved with more interactive, collaborative tools, but felt that the technology might not yet be adequate for what we wanted to achieve.

We were more than pleasantly surprised by the human factors and attractiveness of WebGrid. The free-form HTML documents with embedded widgets in many ways gave us more flexibility than we had experienced in the design of the original RepGrid interface, and interaction with WebGrid has proved natural to our student communities. From a programming perspective, HTML forms provide a cross-platform graphic user interface (GUI) that is simple to prototype, easy to customize, and whose simple primitives help to support the basic human factors guidelines of *uniformity* and *consistency* (Gaines and Shaw, 1984).

One objective in the original WebGrid development was to preserve the stateless nature of the HTTP protocol. No data is stored at the server between transactions. In particular, this makes it possible for us to offer WebGrid as an open service on the web without being concerned about managing the storage of other people's data. However, we are also concerned to support collaborative communities as we have done with RepGridNet (Shaw and Gaines, 1991; Shaw and Gaines, 1993), where members can make their grids available to the community. This raises many problems of authorization, protection, and so on. It is not easy to design a system for the web that has open, easy access, yet avoids data loss or unwanted interference through careless or deliberate actions. WebGrid already supports one user saving their grid locally and linking it to their home page to make it available to others. However, this is an awkward way of supporting an identifiable community, and we are developing a general GroupWeb facility that supports distributed databases of community information, including repertory grid data.

In conclusion, we hope that the example of WebGrid will encourage others to develop interactive collaborative learning systems on World-Wide Web. It is a rich environment for collaboration whose potential we have barely begun to comprehend, let alone tap.

Acknowledgments

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

References

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

Gaines, B.R. and Shaw, M.L.G. (1984). **The Art of Computer Conversation**. Englewood Cliffs, New Jersey, Prentice-Hall.

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. (1993). Basing knowledge acquisition tools in personal construct psychology. **Knowledge Engineering Review 8**(1) 49-85.

Habermas, J. (1968). **Knowledge and Human Interests**. London, Heinemann.

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

Mancuso, J.C. and Shaw, M.L.G., Ed. (1988). **Cognition and Personal structure: Computer Access and Analysis**. New York, Praeger.

Pope, M. and Shaw, M.L.G. (1981). Negotiation in learning. Bonarius, H., Holland, R. and Rosenberg, S., Ed. **Personal Construct Psychology: Recent Advances in Theory and Practice**. pp.157-165. London, UK, MacMillan.

Pope, M.L. and Keen, T.R. (1981). **Personal Construct Psychology and Education**. London, Academic Press.

Rogers, C. (1961). **On Becoming a Person**. London, Constable.

Schön, D.A. (1983). **The Reflective Practitioner**. New York, Basic Books.

Shaw, M.L.G. (1979). Conversational heuristics for eliciting shared understanding. **International Journal of Man-Machine Studies 11** 621-634.

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., Ed. (1981). **Recent Advances in Personal Construct Technology**. London, Academic Press.

Shaw, M.L.G. (1985). Communities of knowledge. Epting, F. and Landfield, A.W., Ed. **Anticipating Personal Construct Psychology**. pp.25-35. Lincoln, Nebraska, University of Nebraska Press.

Shaw, M.L.G. (1994). Women, scholarship and information technology: A post-modern perspective. **Transactions of the Royal Society of Canada 5** 113-131.

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.

Shaw, M.L.G. and Gaines, B.R. (1991). Supporting personal networking through computer networking. **Proceedings of CHI'91: Human Factors in Computing Systems**. pp.437-438. New York, ACM Publications.

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. (1993). Group knowledge elicitation over networks. Bramer, M.A. and Macintosh, A.L., Ed. **Research and Development in Expert Systems X. Proceedings of British Computer Society Expert Systems Conference**. pp.43-62. Cambridge, UK, Cambridge University Press.

Authors Addresses

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

WebGrid may be accessed at:

http://tiger.cpsc.ucalgary.ca/WebGrid/