

WebGrid-III: Access to Distributed Knowledge Modeling Tools through the Web

Brian R Gaines and Mildred L G Shaw

Knowledge Science Institute
University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@ucalgary.ca, mildred@cpsc.ucalgary.ca

Abstract

Organizations are increasingly using the Internet to manage activities through geographically dispersed teams. This is particularly appropriate for knowledge acquisition where input from appropriate experts is vital, and placing constraints of time and location on the knowledge engineering process can undermine effective access. This article reports on WebGrid-III, a system using extended repertory grid and induction tools to offer knowledge modeling through the Internet and standard web clients.

1 Introduction

Developing knowledge-based systems involves knowledge acquisition from a diversity of sources often geographically distributed. The sources include books, papers, manuals, videos of expert performance, transcripts of protocols and interviews, and human and computer interaction with experts. Expert time is usually a scarce resource and experts are often only accessible at different sites, particularly in international projects. The process of building and testing knowledge models across a distributed community would be greatly expedited if wide-area networks could be used to coordinate the activities at different sites. The objective of the development reported in this paper has been to use the web to offer widespread access to distributed knowledge modeling facilities by porting existing knowledge acquisition tools to operate through 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; Gaines and Shaw 1993]. Since grid elicitation tools are used directly by experts it would be useful to have them accessible through any personal computer or workstation with access to the web.

Gaines and Shaw [1997] have described a system WebGrid that gives access to repertory grid elicitation through the web. However, it supports only limited data types and single level inductive modeling, and does not

allow distributed users to collaborate. This paper describes WebGrid-III, a knowledge modeling system accessible through the World Wide Web that supports categorical and numerical data types, multi-level induction for rule-chaining, allows distributed users to compare elicited knowledge structures and is available as a service that can be integrated with other applications.

2 System Architecture

The system architecture is shown in Figure 1. Users communicate with the system through standard web clients, or through an auxiliary knowledge-based system that uses WebGrid-III as a service. The coordinator subsystem receives HTML form data through a common gateway interface to a web server. It generates HTML documents from the processed data using macros embedded in scripts which allows the interaction to be customized for different purposes and languages.

The grid elicitation module offers normal repertory grid elicitation extended beyond rating scales to support ordered and unordered categories, integers, floats and dates. This provides a rich enough representation capability for most case data. It also supports meta-values such as *unknown*, *don't care* and *not applicable*. These are important in allowing the expert to enter stereotypical cases that are prototypes rather than instances, and greatly speed up the knowledge acquisition process.

The cluster analysis module offers the usual repertory grid facilities of hierarchical clustering and principal components analysis. It returns a clickable image, and the visual feedback of relations between cases and attributes enables the user to see how, and on what basis, the cases are differentiated or similar. Clicking on a case or attribute takes the user back into the grid elicitation process to edit that case or attribute.

The meta-values are properly incorporated in clustering by treating them as different types from normal data and defining a metric on cases such the distance between two values, u and v , normalized in the range $[0,1]$ is:-

$$d(u,v) = \begin{cases} 1 & \text{if type}(u) \neq \text{type}(v) \\ |u-v|^n & \text{otherwise} \end{cases}$$

where n is 1 (box-car metric) for hierarchical clustering and 2 (Euclidean metric) for principal components.

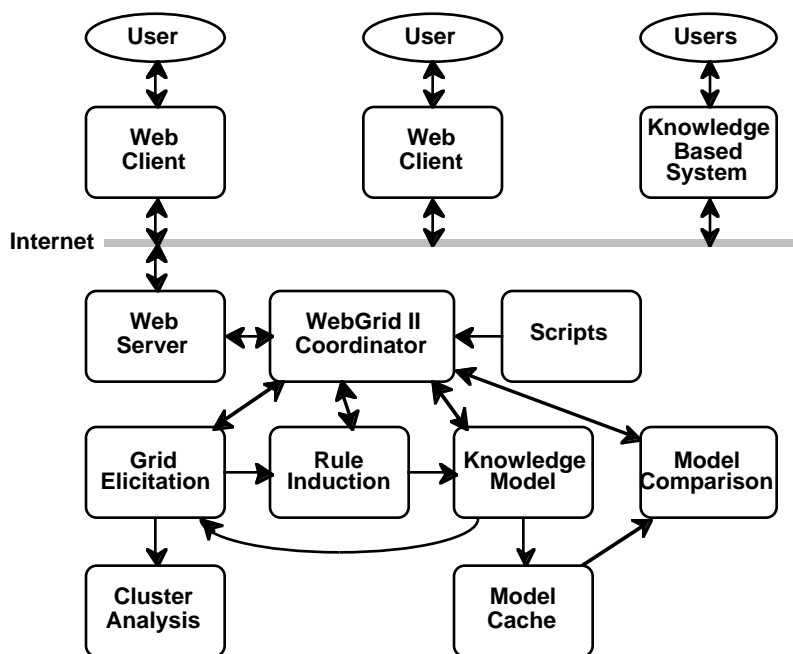


Figure 1 WebGrid-III Architecture

The rule induction module uses a C4.5-like algorithm [Quinlan 1993] to produce modular production rules, or rules with exceptions in the form of Gaines' [1996] EDAG's or Compton and Jansen's [Compton and Jansen 1990] ripple-down rules. It takes proper account of the meta-values in evaluating potential rules by: counting a case as falling under a rule if one or more values relevant to the premise are *unknown* or *don't care*, but only counting that case negatively if the conclusion is incorrect, not positively if it is correct; and counting a case as not falling under a rule if one or more values relevant to the premise are *not applicable*. It also uses a meta-attribute, the *level* of an attribute, entered by the expert or knowledge engineer, to indicate whether one attribute can be used in a premise to infer another. This enables some attributes to be specified as having intermediate values such that rule chains can be induced.

The knowledge model module uses a CLASSIC-like [Borgida, Brachman, McGuinness and Resnick 1989] representation and inference system to enable test cases to be entered and the resulting inferences to be assessed. This is not only used for validation but also as part of elicitation in that, as shown by the loop back from the knowledge model module to the grid elicitation module in Figure 1, test cases with incorrect outcomes can be corrected and posted back as if they were elicited data. This leads to changed rule induction and a corrected knowledge model through a process that is familiar to the expert and does not involve direct editing of the rules or model.

The model comparison module is used to compare the models of different experts to determine if they are using the terminology and attributes in the same way. It uses a model cache subsystem that allows data to be stored

temporarily at the server for purposes of comparison. Data is normally stored at the client to avoid users needing password access to the system, and enable it to be offered widely as an anonymous service on the Internet.

3 Operation

The operation of the system is best understood through an example. Figure 2 shows the initial screen of WebGrid-III. It requests the usual data required to initiate grid elicitation: user name; domain and context; terms for elements and constructs; default rating scale; data types allowed; and a list of initial elements. It also allows the subsequent screens to be customized with an HTML specification of a header and trailer—this capability to include links to multimedia web data is also used to allow annotation, text and pictures, to be attached to elements. The small “?” icons give access to context-sensitive help.

The problem domain is that of “selecting among graduate program applicants” and the intention of the user is to develop a set of rules defining the criteria for selection. The system will incorporate the text typed in by the user into the later dialog, and the user has stated that the elements in the grid are to be called “students” and the constructs “qualities.” The user has set the types of constructs to be “Ratings + Names + Categories + Numbers”—the “Ratings” option elicits a standard repertory grid, the “+Names” option indicates that the user wants to use more advanced facilities for knowledge modeling, the “+Categories” allows categorical qualities to be defined, and the “+Numbers” option allows numeric qualities to be defined. She has left the default rating scale to be 1-5 and has entered the initials of 4 students typifying the range of selection decisions normally made.

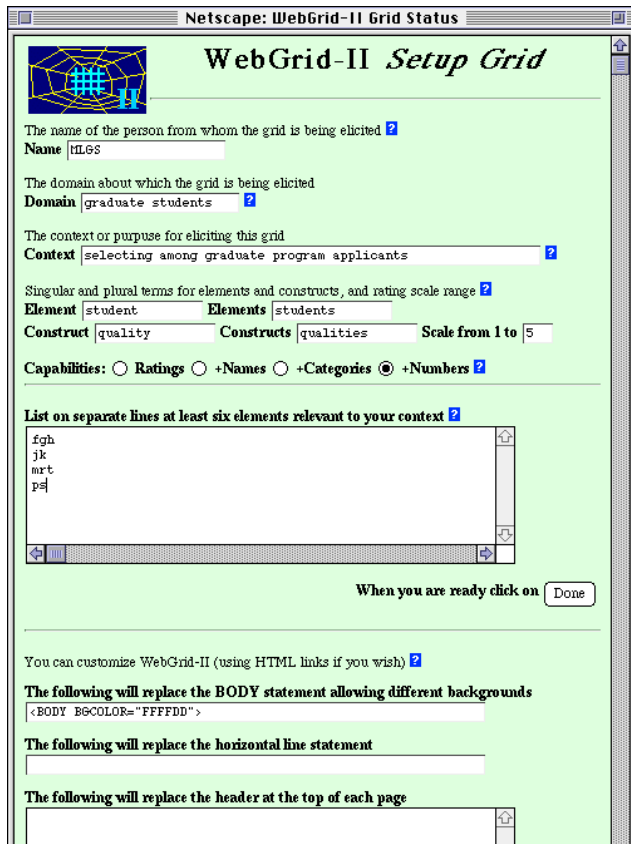


Figure 2 WebGrid-III initial screen

When the user clicks on the “Done” button at the bottom, the browser transmits the data to the remote server which passes it through its common gateway interface to the 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 quality from a triad of students.

The upper part of the screen assumes that a rating scale will be used. The lower part of the screen is generated because advanced features were selected through the “+Numbers” option. It allows the user to vary the rating scale for an individual quality, to select a categorical, integer or float quality, to name the quality, to give it a weight in clustering, a priority level in asking questions in an expert system application, and to specify that the quality is an input for decision making purposes or an output to be anticipated. Since the quality entered is the main decision to be made, to accept or reject a student, and the user wants the system to anticipate the correct decision from the way the student is construed she has made it an output. The user clicks on a radio button to select a student which she construes as different from the other two and enters terms characterizing the quality.

When the user clicks on “Done” the server generates the screen shown on the left of Figure 4 which places a popup menu rating scale alongside each student enabling the user to rate each one along the new quality 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.

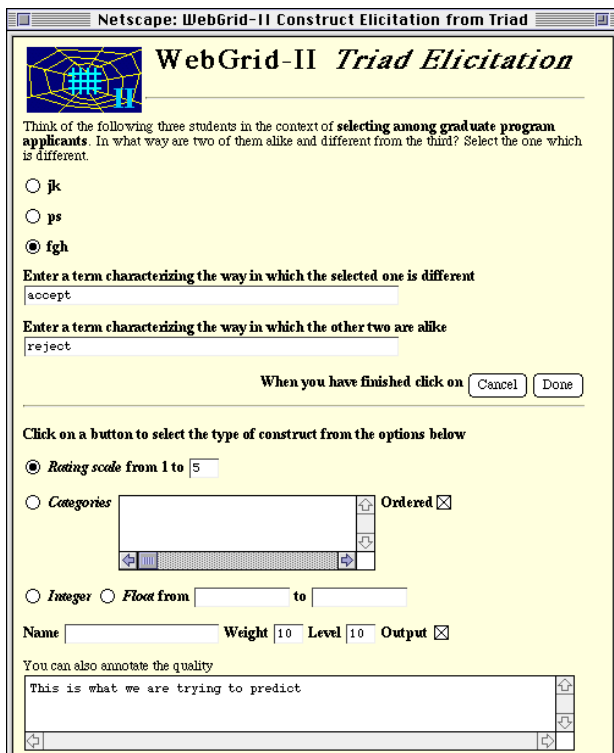


Figure 3 Elicitation from a triad

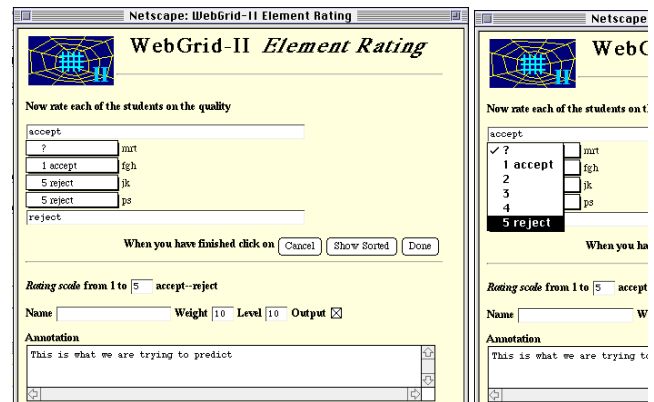


Figure 4 Rating elements

Clicking on the “Done” button in Figure 4 sends the ratings back to the server which generates the status screen shown in Figure 5. This shows the students and qualities entered, allowing them to be selected for deletion, editing and so on. It also offers various suggestions as how to continue the elicitation based on the data entered so far, facilities for analysis, saving the grid, and so on. The user decides to develop another two qualities from triads of elements and enters “high gpa—inadequate gpa” and “good references—inadequate references” since she knows that these are critical to acceptance.

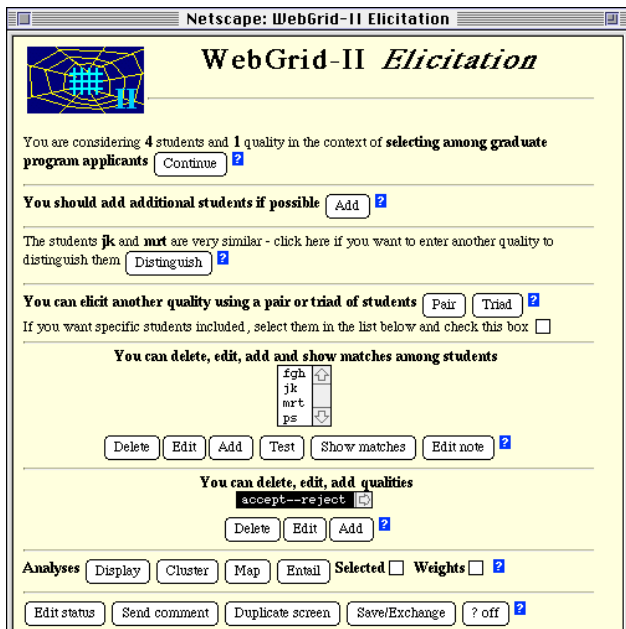


Figure 5 Status screen

The user notes that the students' grade point average (gpa) is important in admission decisions and clicks on "Add" under the list of qualities to add it generating the screen shown in Figure 6.

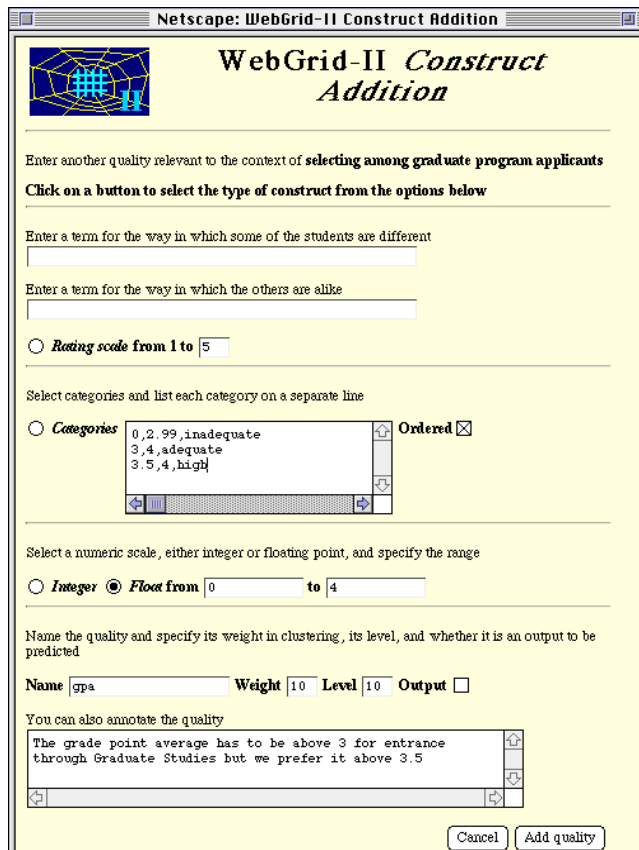


Figure 6 Defining a numeric quality

She fills in the screen shown in Figure 6, entering a quality called "gpa" which is a floating point number between 0 and 4. She uses the "Categories" box to define some categories based on gpa values, that 0 through 2.99 is inadequate, 3 through 4 is adequate and 3.5 through 4 is high. When she clicks on "Add quality" WebGrid-III returns the screen shown in Figure 7 asking her to enter numeric values for the gpa of each student.

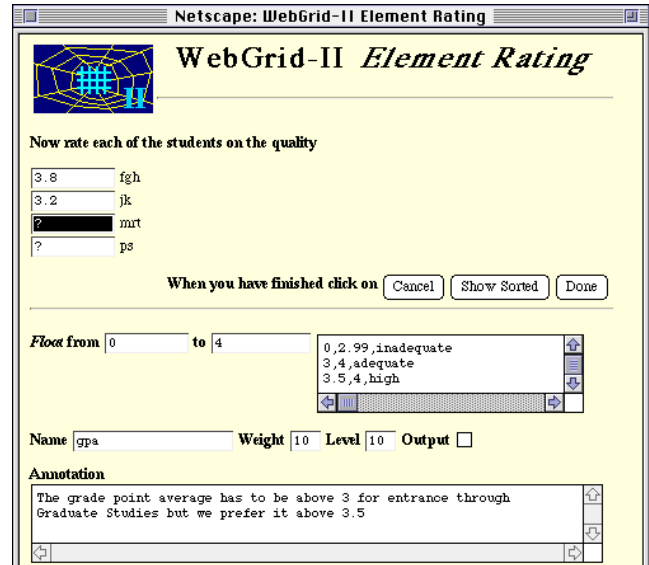


Figure 7 Entering numeric values

She enters another quality, "good references — inadequate references", and decides to run rule entailment to see if the qualities entered account for the admission decisions. This generates the screen of Figure 8 which shows that "ps" might have been expected to be admitted but was not. She realizes this is because he did not have anyone willing to supervise him, enters the quality "willing supervisor—no supervisor", runs rule entailment again and this time gets rules that correctly predict all 4 cases.

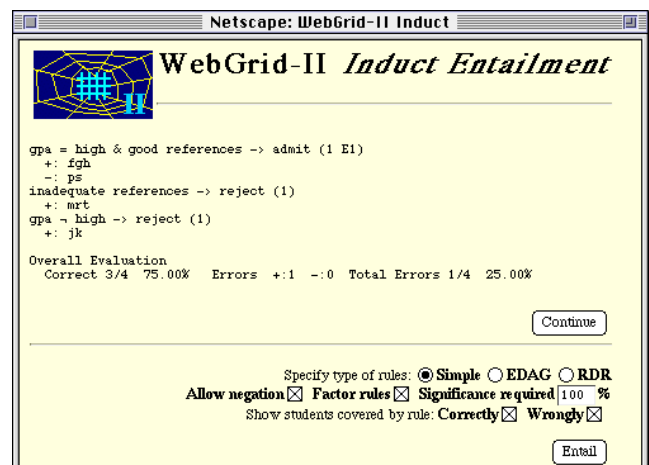


Figure 8 Rules entailed after 3 qualities

She then decides to check the rule against some more test cases and clicks on the “Test” button under the list of students in Figure 5. WebGrid-III generates the screen shown in Figure 9 which enables her to enter data on a test case and infer a conclusion from the rules.

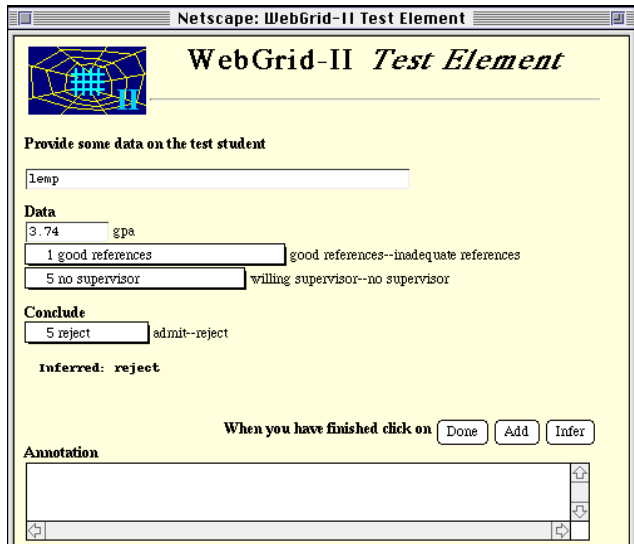


Figure 9 Inference with a test case

When she enters a student “lemp” WebGrid-III infers that he should be rejected but she remembers he was accepted because he had a scholarship and it was expected he would be able to find a supervisor easily. She clicks on “Add” to add this case to the grid and then adds the quality “has scholarship—no scholarship”. She goes on adding students and qualities using the test facility, the suggestions about matches on the main screen, and so on, until she has developed a grid of 12 students and 7 qualities which results in the rules shown in Figure 10.

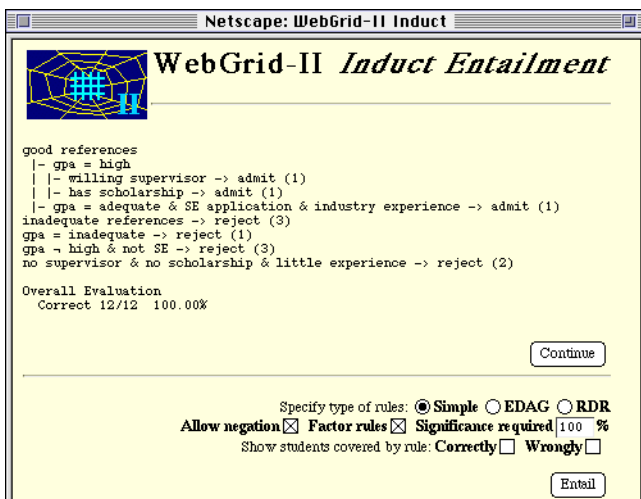


Figure 10 Final rules

The rule for admission is that the candidate must have good references and either a high gpa with a willing supervisor or a scholarship, or must have an adequate gpa

and be applying for the SE specialization which also requires significant industry experience. She decides also to look at rules with exceptions, selects the “EDAG” button in Figure 10 and clicks on “Entail” to generate the screen of Figure 11. This says that a candidate should be rejected unless they have good references and in addition their gpa is high but they do not have no supervisor or no scholarship, or their gpa is adequate, they are applying for the SE specialization and have industrial experience.

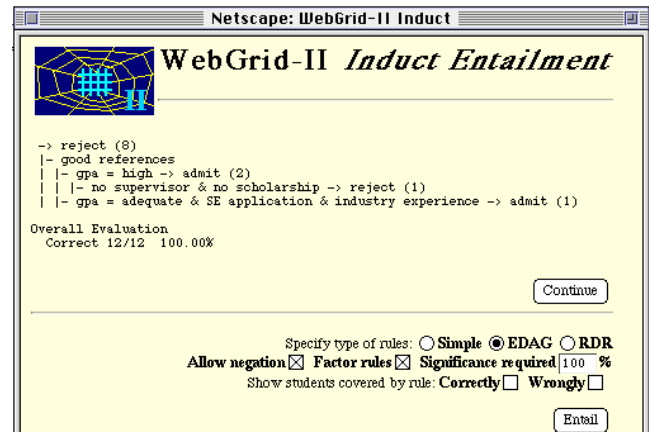


Figure 11 Final rules with exceptions

4 Tests with a Standard Dataset

A useful standard dataset with which to test WebGrid-III’s features is Michie’s [1989] study of NASA’s autolander usage decisions since it has many attributes with *don’t care* values corresponding to the expert’s input of information of relevance through prototypical rather than concrete cases (Fig. 12). The data can be entered as grid of 16 cases with 7 categorical attributes. The error and atmospheric turbulence magnitude (mag) attributes each have 4 ordered categories (ss, mm, lx, xl) and (light, medium, strong, out of range), respectively. This specification induces an inductive bias to test ranges of these attributes.

	auto	vis	stab	error	mag	wind	sign
1	use	no					
2	not	yes	no				
3	not	yes	yes	lx			
4	not	yes	yes	xl			
5	not	yes			out		
6	use	yes	yes	ss	light		
7	use	yes	yes	ss	med		
8	use	yes	yes	ss	strong		
9	not	yes	yes	mm		tail	neg
10	use	yes	yes	mm	light	head	pos
11	use	yes	yes	mm	med	head	pos
12	use	yes	yes	mm	light	tail	pos
13	use	yes	yes	mm	med	tail	pos
14	not	yes	yes	mm	strong	head	pos
15	use	yes	yes	mm	strong	tail	pos
16	use	yes	yes	mm		head	neg

Figure 12 NASA autolander dataset [Michie 1989]

Figure 13 shows the rules generated by WebGrid-III for this data, a default rule to use the autolander with 5 exceptions all specifying visibility and involving clauses such as turbulence is out of range or strong showing the use of the ordered categories. The results are complete and correct, and are a minimal set of rules to model the heterogeneous set of conditions relating to this decision.

```

WebGrid-II Induct Entailment

-> use autolander (9)
|- visibility
| | - errors - ss
| | | - sign = negative -> don't use autolander (2 ?4)
| | | - wind = head & turbulence = out of range | strong -> don't use autolander (1 ?5)
| | | - errors = xl | lx -> don't use autolander (2 ?2)
| | | - turbulence = out of range -> don't use autolander (1 ?5)
| | | - unstable -> don't use autolander (1 ?1)

Overall Evaluation
Correct 16/16 100.00%

Continue

```

Figure 13 Rules with exceptions for NASA dataset

This example, together with that of the previous section, shows how inductive techniques generally used to model large, miscellaneous datasets can also be used to model the small datasets of stereotypical cases generated through repertory grid elicitation.

4 Other Features

One objective of the WebGrid-III design was to support users who wished to collect data through their own methodology, pass them to the server for further elicitation and analysis, and then return data and control to their own system. Passing control is simple since grid data is embedded in hidden fields, and the other system can send an HTTP POST that mimics that of a client. Passing control back is also straightforward since WebGrid II supports user-specified headers and trailers and these can contain JavaScript that returns the data and control of the interaction to another server. It also supports additional hidden fields containing data from the original server which are passed back to it when control returns, thus enabling it to maintain state information.

Tennison and Shadbolt [1998] have used this approach to make WebGrid-III a service integrated with their APECKS (Adaptive Presentation Environment for Collaborative Knowledge Structuring) system on the web. APECKS provides a virtual environment where people who are not in the same place in the real world can meet and talk, and supports a cyclical process of construction, comparison and discussion about knowledge representations through which an organizational memory is gradually constructed.

5 Conclusions

The system described in this paper shows that it is possible to develop comprehensive knowledge modeling systems incorporating a range of artificial intelligence techniques that will operate effectively in a distributed mode through the World Wide Web. WebGrid has been in daily operation

as an open service on the Internet since 1994 and has had over 50,000 different users at sites world-wide. The users range from those with a casual interest in knowledge acquisition techniques, to students in AI courses using the service in projects, through industry-based requirements engineering studies, to research groups integrating the tools within their own systems.

URLs: <http://repgrid.com/WebGrid/>
<http://repgrid.com/reports/>

References

- [Boose 1984] Boose, J.H. 1984. Personal construct theory and the transfer of human expertise. *Proceedings AAAI-84*. pp.27-33. California: AAAI.
- [Boose et al. 1993] 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.
- [Borgida et al. 1989] Borgida, A., Brachman, R.J., McGuinness, D.L. and Resnick, L.A. 1989. CLASSIC: a structural data model for objects. *Proceedings of 1989 SIGMOD Conference on the Management of Data*. pp.58-67. New York: ACM Press.
- [Compton and Jansen 1990] Compton, P. and Jansen, R. 1990. A philosophical basis for knowledge acquisition. *Knowledge Acquisition* 2(3) 241-258.
- [Gaines 1996] Gaines, B.R. 1996. Transforming rules and trees into comprehensible knowledge structures. *Knowledge Discovery in Databases II*. pp.205-226. Cambridge, Massachusetts: AAAI/MIT Press.
- [Gaines and Shaw 1993] Gaines, B.R. and Shaw, M.L.G. 1993. Basing knowledge acquisition tools in personal construct psychology. *Knowledge Engineering Review* 8(1) 49-85.
- [Gaines and Shaw 1997] Gaines, B.R. and Shaw, M.L.G. 1997. Knowledge acquisition, modeling and inference through the World Wide Web. *International Journal of Human-Computer Studies* 46(6) 729-759.
- [Kelly 1955] Kelly, G.A. 1955. *The Psychology of Personal Constructs*. New York: Norton.
- [Michie 1989] Michie, D. 1989. Problems of computer-aided concept formation. *Applications of Expert Systems Volume 2*. pp.310-333. Sydney: Addison-Wesley.
- [Quinlan 1993] Quinlan, J.R., Ed. 1993. *C4.5: Programs for Machine Learning*. San Mateo, Morgan-Kaufman.
- [Rice et al. 1996] Rice, J., Farquhar, A., Piernot, P. and Gruber, T. 1996. Using the web instead of a window system. *Proceedings of CHI'96*. pp.103-117. NY: ACM.
- [Shaw and Gaines 1983] 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. British Computer Society.
- [Tennison and Shadbolt 1998] Tennison, J. and Shadbolt, N.R. 1998. APECKS: a Tool to Support Living Ontologies. *Proceedings of Eleventh Knowledge Acquisition Workshop*. pp.<http://ksi.cpsc.ucalgary.ca/KAW/KAW98/tennison/>.