

An Interactive Knowledge-Based System for Group Problem Solving

MILDRED L. G. SHAW, MEMBER, IEEE

Abstract—A distributed system is described for human-computer interaction based on a network of computers. It aids group problem solving by enabling the participants to share in a construct elicitation process based on repertory grid techniques. Since their development in the context of clinical psychology, these techniques have found a wide range of applications in education, management, and expert systems development. A similar rationale applies: in education the learner is attempting to acquire a specific construct system for the subject matter; in management people with different construct systems are attempting to work together towards common objectives; in expert systems development the knowledge engineer is attempting to make overt and encode the relevant construction system of an expert. The participant construct system described enables a number of individuals to interact through networked personal computers to develop mutual understanding of a problem domain through the use of repertory grid techniques.

I. INTRODUCTION

THE DECLINE in cost of computer hardware in the 1970's has made it possible to provide personal computing through dedicated actual machines rather than timesharing virtual machines as in the previous decade. The only reasons for a user of a personal computer to access another system are for access to greater computing power such as an array processor; access to greater information storage such as high-volume files; and access to information updated remotely such as a centralized data base. The first two requirements, for computing power and storage, are being increasingly satisfied by personal computers. Only the third, access to remotely updated information, has a fundamental systemic component requiring a communication capability. It is this third requirement that the concept of *participant systems* addresses—the interaction of multiple users as an essential requirement of their individual, or mutual, tasks [2], [3].

Fig. 1 shows a typical configuration for a participant system implemented as a network of personal computers. The group of persons who have come together to participate in the system each access it through a personal computer acting as a terminal to a distributed computing network. Terminals to a timeshared central computer could provide an alternative implementation but the delay-free interaction of a user with a personal computer is preferable

Manuscript received June 3, 1987; revised August 27, 1987. This work was supported in part by the Alberta Research Council and in part by the National Sciences and Engineering Research Council of Canada.

The author is with the Department of Computer Science, University of Calgary, Calgary, AB, Canada T2N 1N4.
IEEE Log Number 8821876.

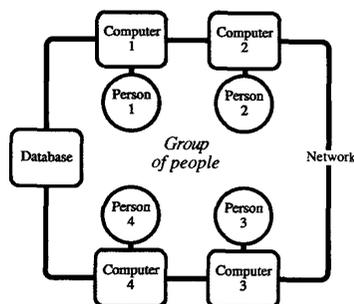


Fig. 1. Typical participant system implementation.

in giving a highly responsive system, particularly for graphic interaction. The data base shown provides storage for information that may be shared among the participants. They may communicate with the data base, or one another, through the network.

Note the emphasis on a group of people; the diagram could be of individuals, each carrying out separate activities on a computer network. However, it is the essence of a participant system that those using it have some joint purpose, giving them coherence as a group.

II. PERSONAL CONSTRUCT THEORY AND THE PERSONAL SCIENTIST

Kelly's personal construct theory [6] provides a systemic model of the cognitive and behavioral processes of the individual, including interpersonal and social interaction. Kelly's concept of the psychology of the person being generated by the requirement to construe experience so as to anticipate the future better is a systemic model with strong mathematical foundations and many implications. In particular, a significant correspondence exists between the modeling process of the person and the expected behavior of a scientist that has led to personal construct psychology being termed that of the *personal scientist* [7]. The scientific process is one in which a descriptive language for data collection is established, data are gathered, a hypothesis is formed, and the hypothesis is tested against further data and revised if it does not conform. The personal psychological process is one in which a descriptive system of personal constructs is established, experience is construed in terms of these constructs, and hy-

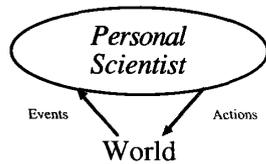


Fig. 2. Personal scientist interacting with world.

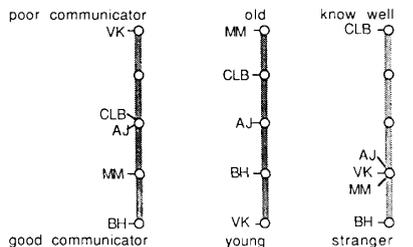


Fig. 3. Repertory grid of managers on three constructs.

potheses are formed as higher level constructs, used to anticipate further experience, and revised if the anticipation proves to be invalid. Fig. 2 shows a personal scientist interacting with the world by construing events, modeling the world, and using the model to generate actions (including predictions) in the world.

A person is modeled as *construing* her experience by dividing it up into chunks (*elements*) and classifying these using a system of bipolar *constructs*. Constructs are themselves construed by the person with other constructs and hence form a hierarchical system. Construing is used to enable a person to anticipate the future as a hypothesized replication of the past. Constructs are changed as a result of errors in anticipation so that personal philosophies are seen as essentially fallible. A person will be more willing to change peripheral constructs rather than core constructs, giving a further order relation on the construct system.

In personal construct theory, learning is seen as the development of the construct system, social processes are seen as the development of shared constructs, and problem solving is seen as the active anticipation of the future. The theory provides complete foundations for human psychological and social processes [11]. Hence it is possible to build technologies based upon it that are very general in concept and application. For example, an expert system in personal construct theory terms is a computer-based system in which the relevant construction system of an expert has been encoded in an active form [16].

In computational terms the elements in personal construct theory may be seen as *entities*, the constructs as *attributes*, and the relationship between elements and entities as *values* in an entity-attribute database. Kelly terms such a data base a *repertory grid*, and Fig. 3 shows how a typical grid may be represented. The elements in this grid are managers represented by their initials. The three constructs illustrate the mix between subjective and objective possible in personal constructs.

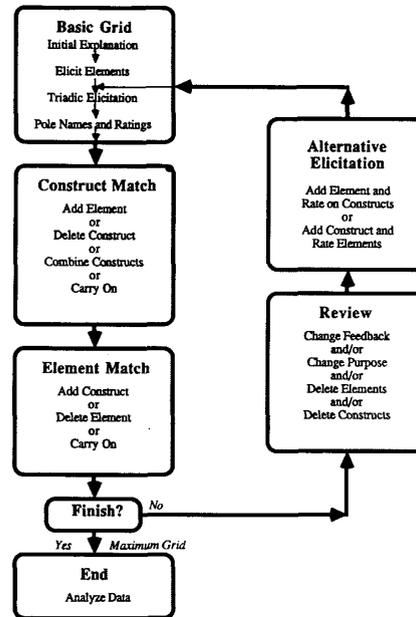


Fig. 4. PEGASUS flow of control.

The manual elicitation of construct systems using structured techniques is tedious, time-consuming, and requires skill, practice, and an appropriate personality. In recent years, as part of the advance in psychological testing technology, computer-based techniques have been developed for on-line elicitation and analysis of construct systems. These have proved very successful and are now in widespread use [7], [8]. Many of these are interactive programs which allow a greater degree of autonomy and more control to be given to the user. PEGASUS [7] is a system for eliciting repertory grids and is part of the PLANET suite of construct elicitation and analysis programs [10]. The commonly used version incorporates continual commentary on patterns in the responses. Six elements are initially chosen by the user with special attention to the purpose for eliciting the grid. The first four constructs are elicited from fixed triads and thereafter random or chosen triads are offered. In the triad method the elements are presented in groups of three—this being the least number which will produce both a similarity and a difference. The subject is asked to say in what way two are alike and thereby different from the third. This is the emergent pole of the construct. The implicit pole may be elicited by the difference method (in what way does the singleton differ from the pair) or by the opposite method (what would be the opposite of the description of the pair).

Real-time processing allows feedback about highly matched constructs and elements. Options offered are to add an element to split highly matched constructs; to replace two highly matched constructs by one; to add a construct to split highly matched elements; to delete one or more elements; to delete one or more constructs; to add a construct without using a triad; to add an element; to

change the level of feedback commentary; and to redefine the purpose for eliciting the grid. The flow of control around these options is shown in Fig. 4.

It would be easy to assume that such interactive programs are merely more convenient ways of elicitation and do not add anything qualitatively new to the process. However, person-computer interaction can have advantages over person-person interaction in some aspects of knowledge processes [15]. There are also certain quite fundamental differences when the elicitation is done in such a way that interpersonal interaction is clearly absent. In particular, when a person is feeding back comments and guidance, it is a natural and ready assumption that the constructs are being injected rather than elicited. It is easy for the subject to believe that the elicited constructs do not come from herself but that a tutorial or debating situation with another person is taking place. In such cases she may well dismiss the construct as irrelevant, and it is necessary to persuade her that this is not so, and the persuasion has to be stronger the more striking and significant the constructs elicited. However, when a computer is the tool by which her construct structure is being reflected or laid bare, then such an assumption of outside injection and interference is far less tenable.

III. PARTICIPANT CONSTRUCT SYSTEM DESIGN

The remainder of this paper describes how the structure of personal construct theory can be combined with the unstructured interaction of a participant system to produce a system for interactively eliciting, exchanging, and comparing repertory grids from multiple simultaneous users on a network of personal computers. The participant construct system (PCS) that does this has been designed as an aid to group problem solving [12], [13]. The users of a PCS are attempting to extend and understand their own thinking and problem-solving capabilities by interacting with other people. They are able to:

- see the relationship of their points of view to those of others;
- explore differing terminology for the same constructs;
- become aware of differing constructs having the same terminology;
- extend their own construct systems with those of others;
- provide others with constructs they have found valuable; and
- explore a problem-solving domain using the full group resources.

One of the most interesting developments of computer-based construct systems has been to attempt to combine the construct systems of a number of people, make them accessible to all, and hence share their knowledge and skills [1], [9], [11], [15]. The extension of interactive construct elicitation techniques to groups exemplifies the operation of a participant system—computer-based com-

munication between multiple users that is essential to the performance of their tasks, rather than merely incidental to their use of a time-shared computer or computer network.

The design, development, and application of participant systems requires an understanding of both the computer technology involved and the psychosocio dynamics of the users. The participant construct system design addresses this requirement by designing a specialist participant system for a domain where large-scale cognitive psychological and sociological theories already exist, implementing it on a network of Apple Macintosh personal computers where an appropriate human-computer interface already exists. The highly structured knowledge elicitation techniques developed for the PLANET system [10], [17] provide a working environment for the users of PCS that avoids the problems of direct unstructured interaction in Cantata [2], [3], [5]. Ultimately, both informal and structured interaction must be combined to allow the full capabilities of participant systems for group problem solving to be exploited.

With the previous discussion in mind, and based on past experience with the PLANET system, the PEGASUS individual grid elicitation procedure was taken as an initial model for PCS. Interaction with PEGASUS, with feedback of construct and element matches, is known from past experience to be a stimulating and enlightening experience; for example, a scientist was surprised to find that he was using the word "time" to label different concepts and was able to clarify his arguments to take this into account. Then to ensure that the group interaction through PCS is seen as important in its own right, the features have been included to

- select additional elements from those entered by other participants;
- compare one's own constructs with those of other participants; and
- exchange complete grids with other participants and evaluate mutual agreement and understanding.

It is often desirable to compare constructs across groups. *Exchange grids* are a way of measuring *understanding* and *agreement* between either two people or one person on two occasions [7]. For example, two people may each elicit a grid in an area of common knowledge. Each may choose his own elements independently of the other and elicit and rate his or her constructs quite separately. Each then makes two copies of the grid, leaving out the rating values. Both these copies are filled in by the other person, one as she uses those constructs on those elements and the other as she thinks the original was completed. By comparing pairs of these grids, it is possible to map the extent of overlap of the agreement and understanding between the two people. Such exchange grid procedures are known to aid individuals in understanding their agreements and disagreements with the way others perceive the world, and to help them understand and use the constructs of others. They give a basis for group problem solving, and hence

PCS may be presented to interested users as a participant system around that group problem-solving procedures may be built.

In choosing a technology for the implementation of PCS the importance of a simple and attractive user interface has been paramount. PEGASUS was designed as a structured interview for users with little or no knowledge of problem-solving techniques. The program largely controls the interaction and limits the user's choices to a few significant decisions. Greater flexibility was both desirable in PCS and also intrinsically required by the overlay of group interaction on the basic construct elicitation procedures.

A poor person-computer interface can completely negate the value of the most advanced participant system. This has been a major consideration in the PCS design and the Apple Macintosh was chosen as a user terminal to PCS because the Macintosh interface encourages an object-oriented programming paradigm in which the user has access to a number of discrete manipulable objects, most of which are potentially accessible at any time. That is, there is very little imposed flow of control. This gives the system tremendous flexibility, allowing much freedom in use and enabling the users to impose their own mode of working upon the system rather than having to conform to some preset sequence.

IV. PCS IN OPERATION

The PEGASUS flow of control in Fig. 3 is replaced on the Macintosh by a set of screens that correspond to the functions in the flowchart, for example, to adding/deleting elements or evaluating matches. This conforms with the Macintosh interface philosophy that the user can activate any one of a number of menus or windows independently of one another. It is also important on the Macintosh that the numeric ratings used in PEGASUS be replaced by a spatial representation of grid data that is readily visualized.

The illustrations of PCS in this section are taken from the *PCS User Manual* [12]. In this example, the user, Bill, is construing *managers* to see which of the ones he knows are *effective* and why. When starting a new topic, the status window must first be checked and changed if necessary. Usually, a new topic will have a specific purpose which should be entered in the appropriate box and kept in mind as the interaction proceeds. The bottom half of the window will change automatically as elements and constructs are added (Fig. 5).

The next thing for Bill to do is to think of some managers and add these as elements. He does this by going to the "elements" window in the "edit" pull-down menu, then typing in the names (or initials) of the managers he would like to think about, one on each line. He adds five (Fig. 6).

As Bill is just starting his grid, he decides to elicit a construct from a triad, so he goes to the "constructs" window in the "edit" pull-down menu and chooses triadic

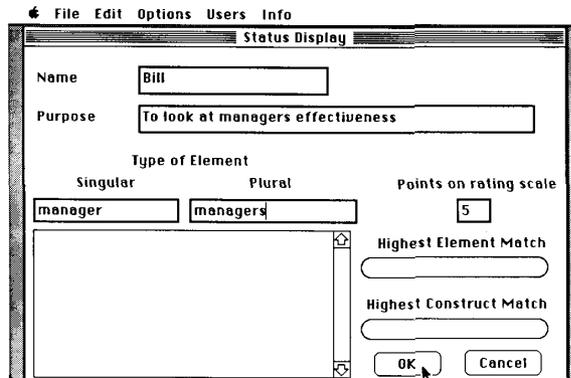


Fig. 5. PCS status window.

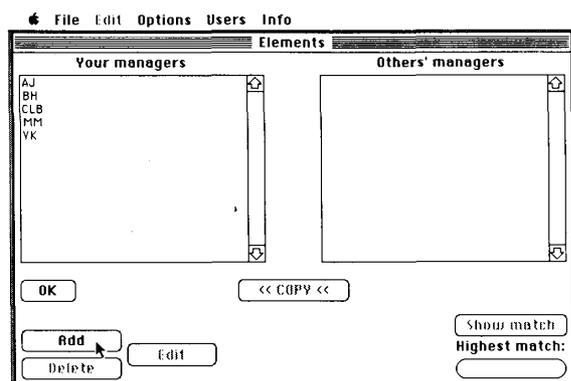


Fig. 6. PCS adding managers.

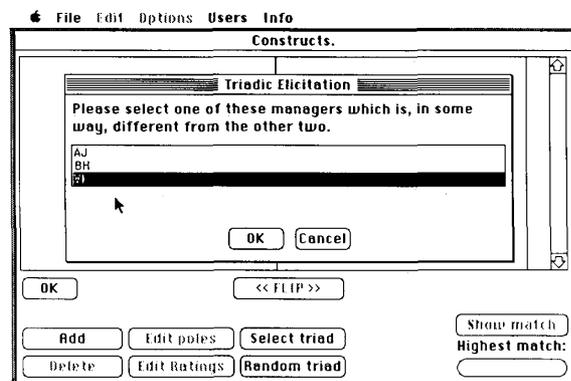


Fig. 7. PCS triadic elicitation.

elicitation from a random triad. Sometimes part-way through the elicitation the user may wish to choose his own triad, otherwise the system will choose one at random. Bill sees the three elements in the triad and decides that VK is the one which is different from AJ and BH (Fig. 7).

Next, Bill has to decide why he thinks AJ and BH are alike and different from VK. He types in the description of VK, "needs supervision," on one pole of the construct

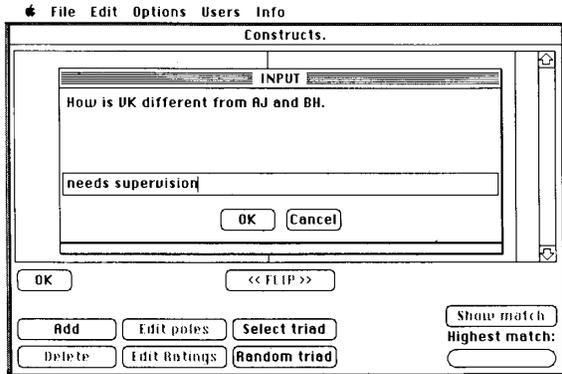


Fig. 8. PCS adding pole names.

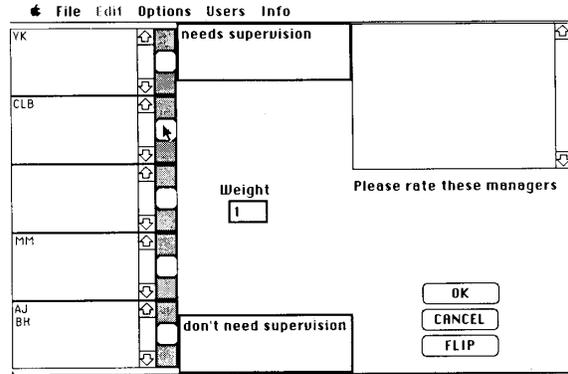


Fig. 9. PCS ratings.

(Fig. 8). Then he names the other end—that is, the way in which AJ and BH are different is that they “don’t need supervision.” This defines Bill’s scale from “needs supervision” to “don’t need supervision”. He now has to rate all his other elements on this scale. If he finds that his original triad AJ, BH, and VK need to be moved to accommodate he others he can do that. In this case, he puts CLB partly down the scale, and MM partly up the scale because MM does not need much supervision and CLB needs a bit more but not as much as VK (Fig. 9).

Now Bill has his first construct. He continues adding several more constructs from triads before he decides to look at who else is on the system. He brings up the “status” window from the “edit” menu and sees that Fred has six elements, and Joe has five. He can also see that he has a high construct match and decides to split it. Perhaps he can use one of Fred’s or Joe’s elements for that (Fig. 10). Bill brings up the “constructs” window from the “edit” menu and clicks on the “highest match” box. (Fig. 11). This takes him to a new screen showing which constructs are matched (Fig. 12). He can see from this screen that most of the time when he is saying that a manager is a “poor communicator,” he is also saying that he has a “narrow view” (VK), and when he is saying that a manager is a “good communicator,” he is also saying that he is “open minded” (BH, MM). AJ is in the middle of the scale in both constructs, and the rating of CLB differs by only one position. This situation could occur because Bill *always* associates managers whom he sees as “poor communicators” with having a “narrow” view, and those whom he sees as “good communicators” as “open minded.” If this is not the case, he can think of another manager, not yet in the element list who is either a “poor communicator” and “open minded” or a “good communicator” and has a “narrow view.”

Bill decides to add a new manager so he brings up the “elements” window from the “edit menu.” Now he can see the managers put in by Fred and Joe in the list of “others’ managers,” and either click on one of those to add it to his list or just type in a new one of his own (Fig. 13). He chooses to add RA from Fred’s and Joe’s lists as he is a “good communicator” but has a quite “narrow view.”

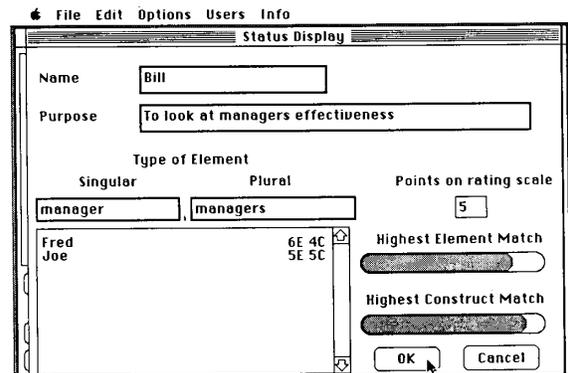


Fig. 10. PCS status window showing matches.

Next he will have to rate RA on each of his constructs in turn. The screen shows his constructs one at a time, and he must place RA where he thinks he fits on each scale (Fig. 14). Note that he may still choose not to rate RA on opposite ends of the two matched constructs, but if he does not, then the constructs will still be highly matched.

Bill continues to elicit his grid with the options available. At any time he can review his purpose on the “status” window to make it more closely reflect his choice of elements and constructs. He can add or delete elements or constructs, use triads, or just type in names and ratings. When he chooses to finish he can close the file from the “file” menu, and it will be stored as part of the total interaction on the participant grid under the topic file name.

There are also two forms of construct “exchange” available to Bill to enable him to compare his perspective on management effectiveness with those of other participants. He can rate a chosen participant’s elements on his constructs either as “Bill would have done” or as “Bill thinks the other participant did.” These two procedures enable him to compare either his *agreement* with the other participant or his *understanding* of the other participant. He can select either of these forms or do both, one after the other.

Bill chose “agreement” with Fred and his first screen was that shown in Fig. 15. He placed these elements on

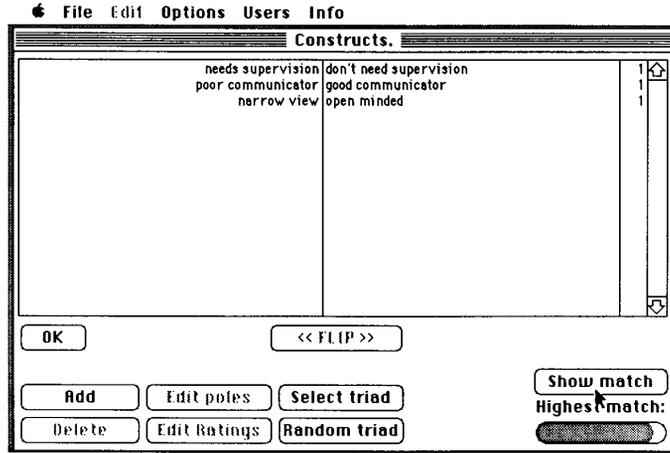


Fig. 11. PCS constructs.

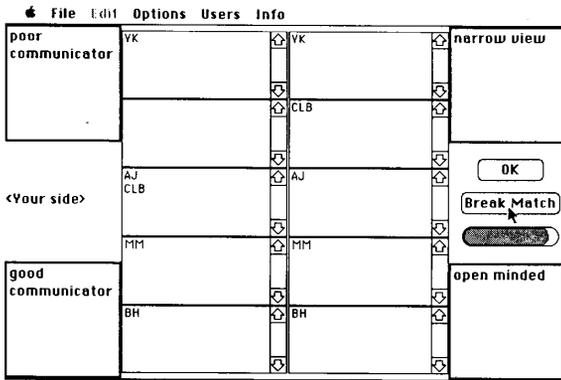


Fig. 12. PCS construct match.

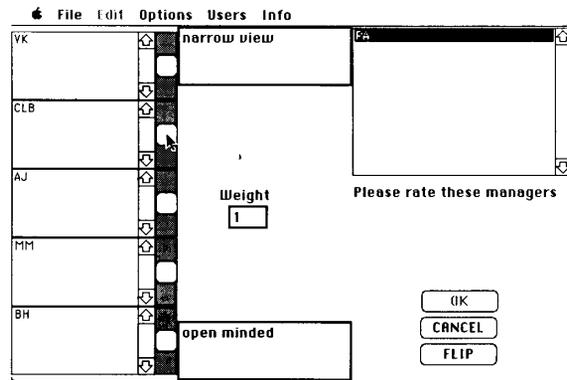


Fig. 14. PCS ratings for new manager.

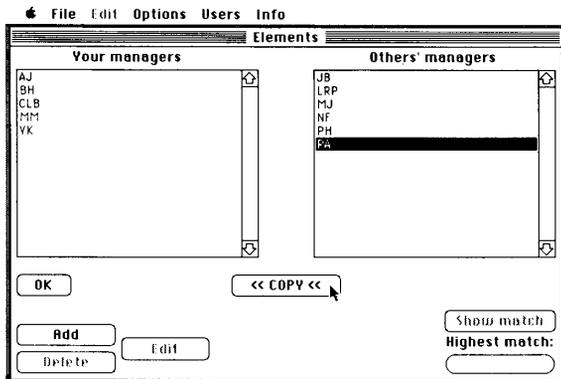


Fig. 13. PCS adding new manager.

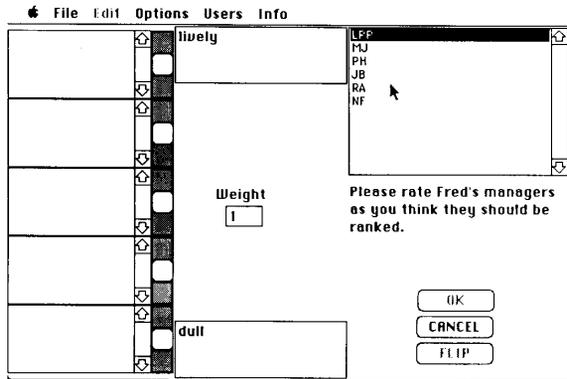


Fig. 15. PCS agreement exchange.

this construct as if it were one of his own constructs, then continued with the next construct until he had done this for all of Fred's constructs.

To repeat this process for "understanding" Fred's constructs, Bill had to go back to the "exchange" menu title, drag to Fred as before, then select "understanding" by clicking on that box. This time he had to rate Fred's

elements on Fred's constructs as he thought Fred had rated them. The screens looked the same for each construct, just the positioning of the elements was different (Fig. 16).

The constructs can then be compared either within a single grid, with one other grid, or with all participants on the system. If the construct names are different, this is done by calculating how similarly the elements are pat-

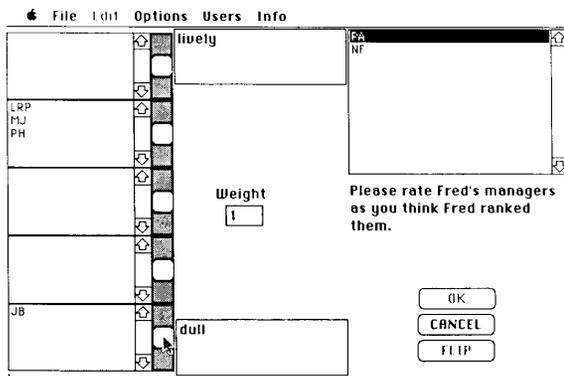


Fig. 16. PCS understanding exchange.

terned on each construct. This is the basis of the focus clustering algorithm used in PLANET that has been explained in detail elsewhere [7], [10]. If the construct names are the same, as in "exchange" grids, then each pair of constructs having the same label, one from each grid, is compared to see to what extent the elements are rated in the same places. Fig. 17 shows the degree of similarity in each construct represented in a bar. It can be seen that Bill totally agrees with Fred on which managers have "been here a long time" or are "new to the job." They mostly agree on which managers "stick to the rules" or "skate round the rules," and "work long hours" or "finish early." However, there are varying degrees of disagreement on the other constructs, and almost complete disagreement on which managers "know more than I do" or are "not very knowledgeable."

Bill can click on any bar to see how the managers are placed on the construct both by himself and by Fred. He will see a screen similar to that in Fig. 12. He decides to look at the last one, where they disagree the most, and can see that he and Fred disagree because they have different specialisms, and Fred has chosen to construe managers in his own field that is highly technical, whereas Bill is in a more general management field and their technical knowledge is irrelevant to his job. It is obvious why he and Fred disagree once he realizes what has happened; Bill would be in the wrong job if it were not the case. Care must be taken with subjective constructs such as this one that are related only to an individual viewpoint.

V. RELATED WORK

The PCS is still under development at Alberta Research Council. Several other systems have been designed, implemented, and evaluated in different domains. PLANET has been widely distributed internationally for over seven years and used in educational, clinical, and management studies; all the other systems mentioned here originate from the ideas in planet. It runs on the Apple II family of computers. ETS, Aquinas, KITTEN, and KSS0 are rapid prototyping systems that use repertory grid methods for eliciting knowledge from experts. ETS and Aquinas were developed at the Boeing AI Center [1] and run on a Xerox

Lisp Machine. KITTEN [19] runs on a network of Apollo workstations, and KSS0 [4] on a network of Macintosh computers. They all use repertory grid methods for eliciting knowledge from experts, and other knowledge acquisition techniques, and have various minor advantages and disadvantages over each other. KITTEN and KSS0 use direct manipulation techniques similar to those in PCS. All these systems are research tools and, as such, are continually under development and modification. RepGrid [14] is a commercially available version of KSS0 and is only concerned with grid elicitation, exchange, and analysis, omitting all the other knowledge acquisition and rule building techniques used in the knowledge acquisition systems.

A number of studies are currently in progress to evaluate the technique for the elicitation of expert knowledge and problem solving in the oil and gas industry [20]. One study is concerned with the evaluation of spatial interpolation techniques for mapping surface structure, and another with troubleshooting in oil and gas pipeline valve maintenance. It is hoped to show how expertise is elicited from several experts, combined by using exchange grid procedures, and compared with a text analysis of expert literature on the topic.

VI. CONCLUSION

A participant system is defined as a computer-based communication system for multiple users that is essential to the performance of their mutual task, rather than merely incidental to their use of computer network. Personal construct theory can be seen as a theory of cognitive science that has deep systemic foundations, models the person as an anticipatory system, and has been operationalized through computer programs that have proved very powerful in expert system development [18].

The new technology of interactive networked personal computers make it feasible to develop systems for group problem-solving participant systems. This paper has described such a system based on personal construct theory. The participant construct system interactively elicits, exchanges, and compares repertory grids from multiple simultaneous users on a network of Macintosh computers. The users of the system can extend and understand their own thinking and problem-solving capabilities by interacting with other people to see the relationship of their points of view to those of others. They can also explore differing terminology for the same constructs; become aware of differing constructs having the same terminology; extend their own construct systems with those of others; and provide others with constructs they have found valuable. By making explicit the areas of agreement and disagreement between all pairs of people in the group, areas of commonality can be identified, and hence assumed by the group. Group resources can be used to examine and explicate the unique viewpoints (constructs) to decide if it would be advantageous to consider them more generally, as in a brainstorming evaluation session. This leads to a very rapid focusing of ideas and perspectives in a

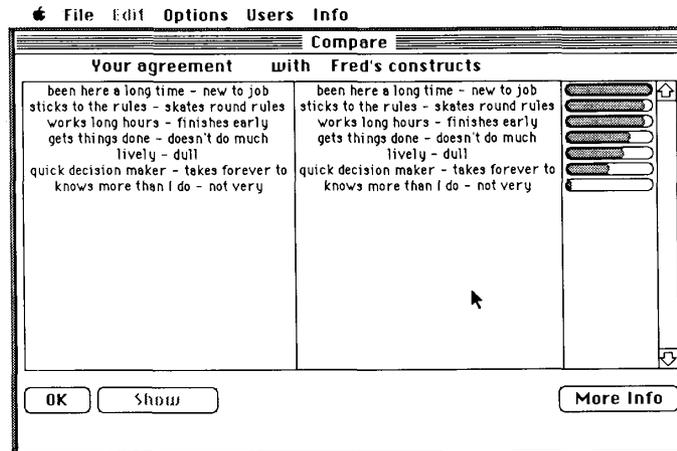


Fig. 17. PCS comparing agreement with Fred.

problem-solving domain. Thus PCS enables a group of people to explore a problem-solving domain using the full group resources.

ACKNOWLEDGMENT

The author would like to thank the reviewers of this paper for their helpful comments, and Dr. B. R. Gaines who has greatly contributed to this work.

REFERENCES

- [1] J. H. Boose, "Rapid acquisition and combination of knowledge from multiple experts in the same domain," *Future Comput. Syst.*, vol. 1, pp. 191-216, 1986.
- [2] E. Chang, "Participant systems," *Future Comput. Syst.*, vol. 1, pp. 253-270, 1986.
- [3] —, "Participant systems: Group human-computer interaction," in *Proc. 1986 Int. Conf. Syst., Man Cybern.*, Atlanta, GA, 1986, pp. 1337-1342.
- [4] B. R. Gaines and M. L. G. Shaw, "Knowledge support systems," in *Proc. ACM MCC—Univ. Research Symp.*, Austin, TX, 1987, pp. 47-66.
- [5] R. Kasperski, L. Mellen, and E. Chang, CANTATA: Group protocols in a conferencing environment," in *Proc. 1986 Int. Conf. Syst., Man Cybern.* Atlanta, GA, 1986, pp. 1343-1346.
- [6] G. A. Kelly, *The Psychology of Personal Constructs*. New York: Norton, 1955.
- [7] M. L. G. Shaw, *On Becoming a Personal Scientist*. London: Academic, 1980.
- [8] —, Ed., *Recent Advances in Personal Construct Technology*. London: Academic, 1981.
- [9] —, "Conversational heuristics for eliciting shared understanding," in *Recent Advances in Personal Construct Technology*, M. L. G. Shaw, Ed. London: Academic, 1981, pp. 31-44.
- [10] —, "PLANET: Some experience in creating an integrated system for repertory grid applications on a microcomputer," *Int. J. Man-Machine Studies*, vol. 17, pp. 345-360, 1982.
- [11] —, "Communities of knowledge," in *Anticipating Personal Construct Psychology*, F. Epting and A. W. Landfield, Eds. Lincoln, NB: Univ. of Nebraska Press, 1985, pp. 25-35.
- [12] —, *Participant Construct System User Manual*. Calgary, AB, Canada: Alberta Research Council, ARC-CPCS-PCS-MANUAL: FEB 86, 1986.
- [13] —, "PCS: A knowledge-based interactive system for group problem-solving," in *Proc. 1986 Int. Conf. Systems, Man, and Cybernetics*, Atlanta, GA: 1986.
- [14] —, "Interactive elicitation and exchange of knowledge in group problem solving," *Int. J. Personal Construct Psychology*, to appear, 1988.
- [15] M. L. G. Shaw and B. R. Gaines, "The personal scientist in the community of science," in *General Systems Research and Design: Precursors and Futures*, W. J. Reckmeyer, Ed. Soc. for General Systems Research, 1981, pp. 59-68.
- [16] —, "A computer aid to knowledge engineering," in *Proc. British Computer Soc. Conf. Expert Syst.*, Cambridge, MA, 1983, pp. 263-271.
- [17] —, "Interactive elicitation of knowledge from experts," *Future Comput. Syst.*, vol. 1, pp. 151-190, 1986.
- [18] —, "A framework for knowledge-based systems unifying expert systems and simulation," in *Intelligent Simulation Environments*, P. A. Luker and H. H. Adelsberger, Eds. La Jolla, CA: Soc. for Computer Simulation, 1986, pp. 38-43.
- [19] —, "KITTEEN: Knowledge initiation & transfer tools for experts & novices," *Int. J. Man-Machine Studies*, vol. 27, pp. 251-280, 1987.
- [20] M. L. G. Shaw and J. B. Woodward, "Validation of a knowledge support system: Construing and consistency with multiple exports," *Int. J. Man-Machine Studies*, vol. 28, 1988, in press.



Mildred L. G. Shaw (M'87) received the B.Sc. degree in mathematics and the M.Sc. degree in computer science in 1971 and 1972, respectively, from the University of London, UK, and the Ph.D. degree in 1978 from Brunel University, UK.

In 1972-1982 she was a Senior Lecturer at Middlesex Polytechnic, UK, and from 1982 to 1985 Professor of Computer Science and a member of the Faculty of Graduate Studies in Psychology at York University in Toronto, ON, Canada. She is currently a Professor of Computer Science at the University of Calgary, Calgary, AB, Canada. Her research interests are mainly in the area of cognitive science, including the developments in and applications of personal construct theory; automation of repertory grid techniques and analysis; the study and improvement of human-computer interaction; knowledge engineering for expert systems; and the system-theoretic modelling of the interaction between personal models of the world and social determinants.

Dr. Shaw is a Fellow of the Institute of Mathematics and its Applications, a Fellow of the British Computer Society, an Associate of the British Psychological Society, a Member of the American Psychological Association, and a Member of the Cognitive Science Society. She is the Managing Editor of the *International Journal of Future Computing Systems*, published by Oxford University Press and Maruzen (Japan). She is the author or editor of five books on aspects of computing and psychology and has published a wide range of papers on human factors in computing, over 80 in refereed journals, books, and conference proceedings.