INTERACTIVE COMPUTER PROGRAMS FOR ELICITING PERSONAL MODELS OF THE WORLD

MILDRED L. G. SHAW

Centre for the Study of Human Learning,
Brunel University, Kingston Lane,
Uxbridge, Middlesex.

The Centre for the Study of Human Learning is interested in encouraging self-organisation in learning by helping people to investigate, expand and rebuild models for construing which will enable them to be more successful learners and users of experience. This paper describes how conversational methods are used which are content free, and which lend themselves superbly well to the real-time data processing of a computer. The application of these model building facilities has been in areas such as learning skills; psychotherapy and 'becoming'; management selection and development; industrial inspection and quality control; art and architecture; the maintenance of electronic equipment; career guidance and the training of counsellors; and in the education of both children and teachers.

Conversations may take place between two people, in a group of people, or within one person such as Perls' (1969) 'top dog' and 'underdog', or Pask's (1973) 'P Individuals'. Conversational heuristics have been embodied in content free computer programs which have the capacity to encourage and control conversation as rigorously and systematically as traditional experimental methods are monitored and controlled. In this context the Centre can be seen as a psychological tool-making unit.

The Programs

The repertory grid is used as a conversational tool to help people to become more aware of the patterns of thought and feeling implicit in their responses. The FOCUS program takes a completed repertory grid and re-orders it for talk-back purposes. The elements and constructs are sorted in such a way as to highlight the pattern of responses in terms of the similarities and differences. FOCUS is described later in more detail.

PEGASUS is an interactive program which elicits a grid using a conversational heuristic. Feedback commentary is given immediately the responses are entered. Again, it will be described in detail later.

SOCIOGRIDS is a method for examining the commonality of construing in a small group. The area of interest is represented by the shared elements, and from each individually elicited grid a pattern is constructed showing the subgroups of people who are construing in the same manner, and the content of construing which leads to these patterns.

ARGUS elicits six grids simultaneously from one person from different points of view, which are then processed on SOCIOGRIDS to explore the relationships of these viewpoints.

CORE examines the change between two grids with the same element and construct names. Details are given in 'Notes on Computer Programs' (Shaw 1977).

FOCUS - Grid Analysis for Feedback

The traditional methods of grid analysis have been factor analysis and principal component analysis. Later, multidimensional scaling was used, and more recently methods of cluster analysis. The method developed and used by the Centre is called FOCUS and is a type of non-inclusive two-way cluster analysis (Thomas and Shaw 1976).

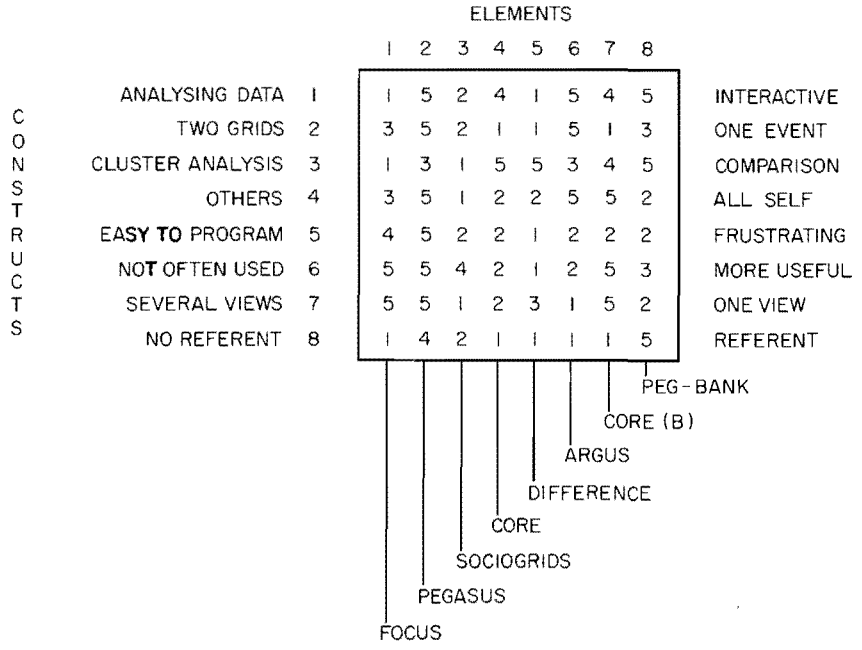

Fig. 1. Raw grid showing the construing of the programs.

The method was devised mainly for use in feeding back

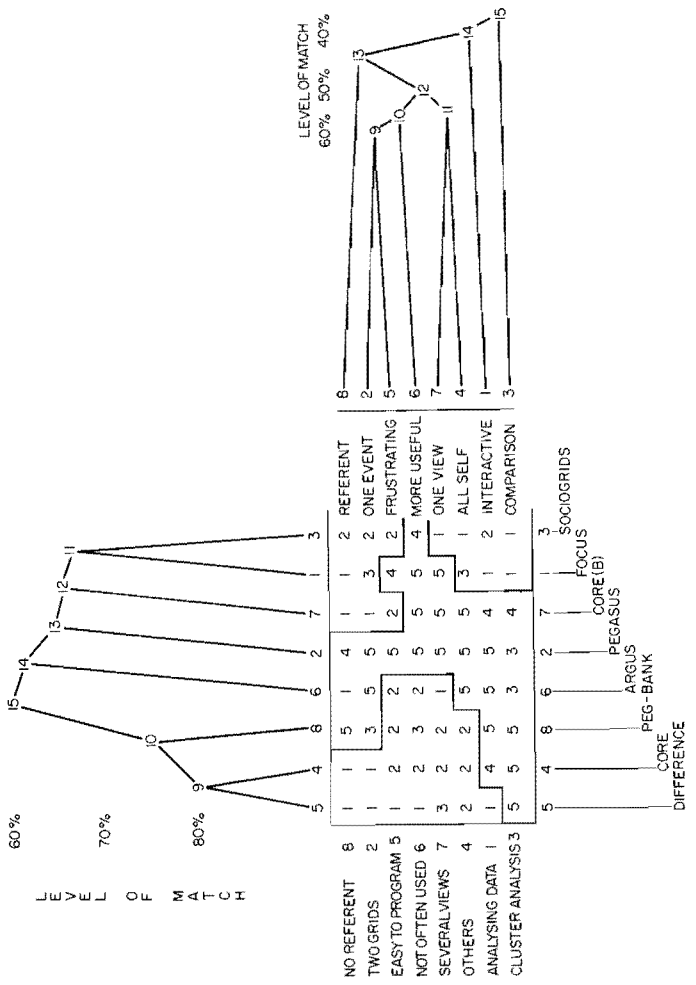

Fig. 2. The focused grid with the element and construct trees.

the analysis of the grid to the subject without displaying any mathematical 'magic', complex computer output, or general problems of naming factors or components. The focused grid retains the raw data, (shown in Figure 1) but presents it in a re-ordered form with tree diagrams indicating how the re-ordering was derived (shown in Figure 2).

This is an example of a focused grid elicited using the PEGASUS program. (For purposes of explanation the elements are in fact the programs discussed in this paper as construed by the author using a 5-point scale.) The element tree above the grid is examined first. Initially each element is considered as a cluster of one, and the first combination to be formed is cluster 9 which contains element 5, DIFFERENCE and element 4, CORE. Clearly these are similar on all constructs but construct 1: they have no referent, use two grids, were easy to program, are not often used, incorporate several views, concern others rather than just self, and are comparisons rather than cluster analysis. PEGASUS-BANK then joins CORE to make cluster 10. Again the differences are only on one or two constructs. Looking down the patterns of ratings, element 2 (PEGASUS) is most different from DIFFERENCE, and the progressive change in patterning across the grid shows that SOCIOGRIDS on the right hand side is quite similar to DIFFERENCE on the left. So that, if the constraint of the linear re-ordering was not necessary, and another dimension could be allowed in the representation, the ends of the grid would swing round the back and come quite close to each other.

Looking now at the constructs, the first peculiarity is that none are highly matched. In fact clusters 9, 10, 11 and 12 are all happening at almost the same level. The types of construct also vary. It seems that 3, 1, 4 and 7 are descriptive, 6, 5 and 8 are subjective and 2 is not very meaningful. Some indicate the point of view of 'me as user', and others of 'me as programmer'. Thus Figure 2 shows how two-way clustering re-orders the grid responses for easier and more meaningful feedback. This is the essence of the FOCUS technique. The majority of our other programs use the FOCUS algorithm as a base.

PEGASUS - The Grid as a Cognitive Mirror

Used in ways similar to this, the grid is acting as a cognitive mirror. It is an empty, content-free structure which reflects back to the user himself, his models of the world, and his constructions on the world. In the PEGASUS grid elicitation procedure, real-time data processing by the computer allows the feedback to be immediate and interactive. In practice, the computer terminal takes the place of the psychologist/therapist in the elicitation procedure.

Kelly's view of 'Man as Scientist', or as it is now being described 'Personal Scientist', shows man as modelling reality in order to anticipate and act on the basis of this anticipation. The quality of a person's models will determine the level of skill, coping, competence and creativity he will be able to achieve, and the PEGASUS grid is a useful tool for heightening awareness of the world. On the one hand it can be used in a 'grid-centred' way, that is as a grid elicitation package with interactive feedback during the elicitation and analysis of the results on completion of the grid. On the other hand it can be used in a 'learning-centred' way. Learning must necessarily involve changes in construing, and PEGASUS encourages the user to review and revise his model as he becomes able to differentiate in ways he previously was not doing, and hence become more able to learn from experience. By giving the learner continual feedback when constructs and/or elements are being used in a very similar way, the computer is doing what few human beings can do with any degree of accuracy.

Figure 3 is a user's flow diagram of PEGASUS (Thomas and Shaw 1977). It does not demonstrate the flow of the program but only the interaction between the human user and the computer. It is divided into six sections. The first one is the 'Basic Grid' in which explanations are given and the first four constructs are elicited. The choice of elements largely determines the depth of interaction that can be achieved. The elements must relate to the purpose the user had in mind, and represent as fully as possible the universe of discourse which is to be explored. As each construct is elicited, the poles are named and ratings assigned to each of the elements on this dimension. The elements are then grouped according to the ratings given to highlight the patterning and allow the user to revise his ratings or pole names if he wishes.

The second section 'Construct Match' provides feedback when two constructs are being used similarly. The options given are to add an element to split these highly matched constructs; to delete a construct if the user feels that one subsumes the other; to combine the two constructs into one; or to continue leaving both constructs in if he feels that they are contributing differently to his grid. In the earlier example of the grid in Figure 2, the on-line feedback led me to split the constructs 'recent - long standing' and 'not often used - more useful' by adding the element PEGASUS-BANK. Later, however, the two constructs 'recent - long standing' and 'easy to program - frustrating' were highly matched, and I chose to delete 'recent - long standing' at that stage.

The third section is 'Element Match' which gives feedback on elements which are being construed similarly. The options given here are to add a construct which puts one of the elements on the left pole and the other on the right pole; to delete one of the elements, which may or

may not be appropriate depending on the type of element being used; or to carry on leaving both elements as they are.

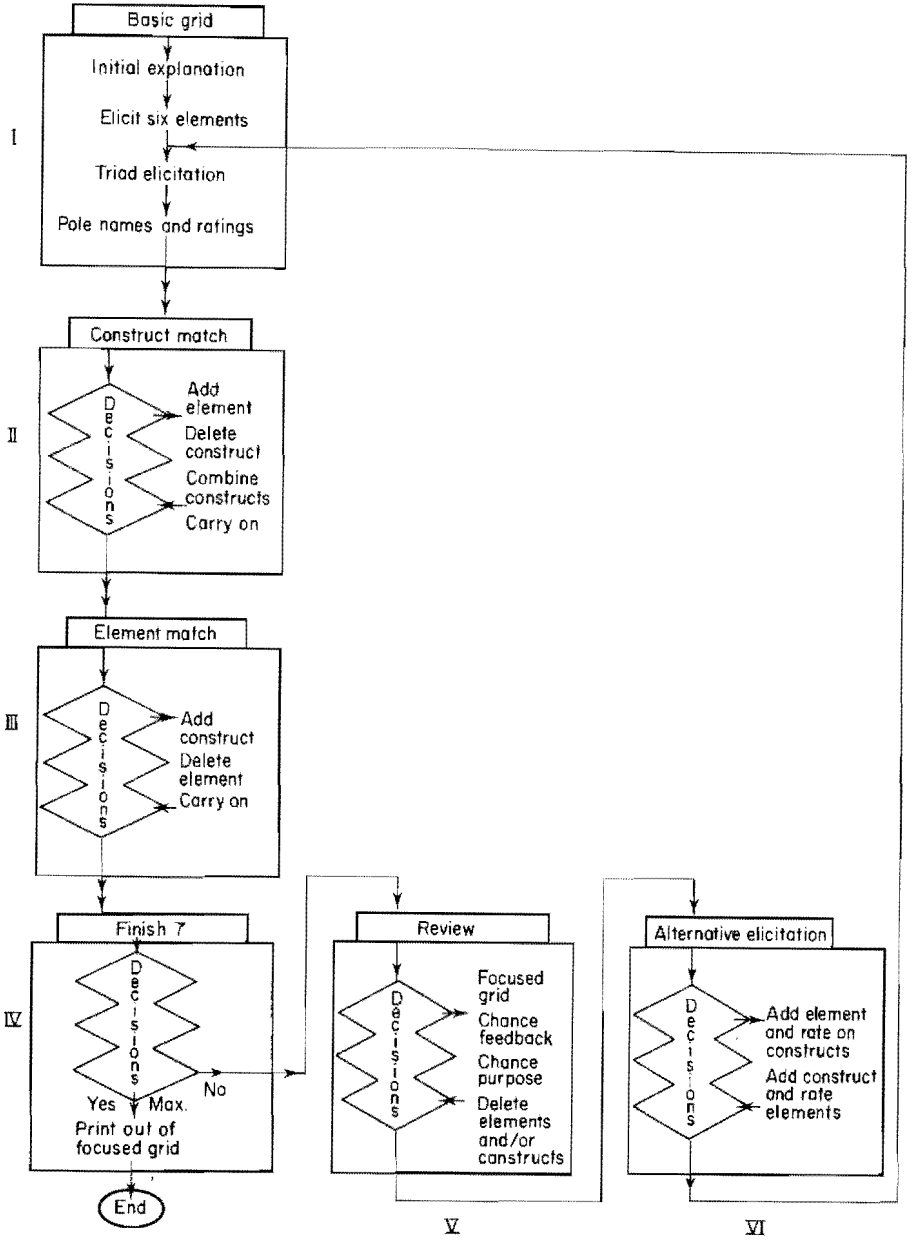

Fig. 3. Flow diagram for the PEGASUS interaction.

Since elements are less often deleted than constructs when a high match occurs, the resulting grid usually displays more highly differentiated constructs than elements as in Figure 2. If a construct or element has been added, the ratings of the other elements on that construct, or that element on all the constructs, must be entered.

Section four is 'Finish?'. This gives the user the opportunity to complete his grid at this stage during the elicitation. If he chooses to do so he is offered a choice of printout of either the results of the analysis or all the analysis. If at this stage the grid is maximum size the user must finish, but if he has the maximum number of elements he may add constructs to bring it to the maximum size.

The fifth section, 'Review', allows the user to see his grid in focused form. He may also adjust the intensity of feedback, and review or refine his purpose for eliciting the grid. As the elicitation proceeds the addition of elements and constructs may shift the boundary of the universe of discourse, and the purpose may need to be modified. The user may also wish to delete elements or constructs which he feels to be outside the boundary of his grid.

In section six 'Alternative Elicitation', the user is given more freedom to add an element, or a construct without using a triad. In the previous example, as the elicitation proceeded it became clear that CORE was being construed in apparently inconsistent ways. This was due to the fact that there are two main uses for this program: for comparing two grids done by the same person on two separate occasions, and for investigating the shared understanding between two different people. The problem was solved by splitting the element CORE into two elements - CORE being the version for two people and CORE (B) the version for one person over time. In the analysis, if these are in fact being used in the same way they will be highly clustered, which is not so in this case.

The user is also invited to choose his own triad for eliciting a construct, but if he chooses not to do this a pseudo-random number routine is used rather than fixed triads.

This is the basic structure of PEGASUS. An alternative form is MIN-PEGASUS which allows constructs to be added or deleted but does not give feedback commentary on matches between them. This version is used when one wants to discover how the user is construing in the situation rather than pushing him to differentiate highly matched elements and constructs. PRE-PEGASUS allows the user to start his grid on one occasion and continue or complete it at a later date.

PEGASUS-BANK is an addition to the PEGASUS program. It allows a bank of constructs to be stored in the computer

representing an 'expert' view of an area of public knowledge. As the processing takes place, continual comparison with the bank gives feedback on how the user's constructs map on to the expert's construing of the same elements. Since the comparison is made in terms of how the construct orders the elements rather than in terms of the verbal labels, it is often found that although a person may have only a vague idea of the expert's terms, he may in fact be using very similar constructs. One example is that of a grid using animals as elements. The biologist had elicited a grid which was stored in the bank, the user had elicited a construct which he called: 'horrible creepy crawlies - nice, soft cuddly ones'. The computer's feedback response was that 'horrible creepy crawlies' was highly matched with the biologist's term 'arachnida', and 'nice, soft cuddly ones' was being called 'warm-blooded mammals'. Very often the user is both surprised and enlightened to find the similarity between the patterning in his grid and that of the expert. This technique therefore provides a sound basis for assessment and a useful starting point for training.

SOCIOGRIDS is a program which analyses the results of common experience and/or training in a small group (Thomas, McKnight and Shaw 1976). The negotiated elements represent the subject under discussion and each person elicits his own grid using personal constructs. The constructs from each person represent similar or different thoughts and feelings on the subject. Any pair of grids can be matched one with the other to obtain a measure of overlap or commonality of construing. The FOCUS algorithm provides a simple method for doing this. All possible pairs in the group are analysed in this way and a series of sociometric-like diagrams are drawn, designated 'socionets', to illustrate patterns of similarity and dissimilarity of construing within the group. In addition, all the constructs from each individual are classified in terms of the extent to which they are shared. A 'mode grid' is constructed from those constructs most frequently used by members of the group and is used as a common referent with which each individual is compared.

The Delphi technique used in conjunction with SOCIOGRIDS makes use of the mode grid to allow each individual in the group to clarify his thoughts and feelings in the light of the constructs most used by the group, and to revise his constructs to highlight his position in the group if he wishes to do so.

The ARGUS program describes a conversation between several roles or points of view within one person. The user is asked to name six people who are central to him in the area under consideration, and he is then asked for three constructs about these people. Taking the point of view of one of his elements he is asked to re-rate all

the elements on the existing constructs and add a new construct which he thinks would be important for that person. This is done from the position of each element in turn. As the elicitations continue, he builds up six grids each with the same element and construct labels, but with different ratings in the grids. If the elements represent a set of significant others, each of the six grids captures an important personal perspective for the elicitee. These six grids are then processed on SOCIOGRIDS to investigate how these personal perspectives relate one to another.

The CORE program compares two grids in which the same constructs and elements are used, to identify the stable or common component. It does this interactively, allowing the user to delete alternately the element which is construed most differently and the construct which is used least similarly, until he decides to stop. The remaining stable elements and constructs constitute the 'core grid'. One application is that of 'exchange' grids used to explore the extent of understanding and/or agreement between two people; another is to chart change in one person over time, to assess the effectiveness of therapy or training for example.

In conclusion, these programs offer a facility that turns the repertory grid back into a useful and developing technology. Other projects include the possibility of extending the technology of learning-centred structures to non-grid techniques. These might include different ways of discovering personal constructs at different levels of organisation; and alternative ways of representation such as Venn and Carroll diagrams from mathematics, and hierarchies, heterarchies, networks, trees and linked lists from computer science.