3 PLANET: A Computer-Based System for Personal Learning, Analysis, Negotiation, and Elicitation Techniques

Mildred L. G. Shaw and
Brian R. Gaines

Repertory grids have proved to be powerful tools in a wide range of disciplines and applications. However, elicitation by people requires skills in interviewing techniques, personal construct psychology, and the avoidance of interpersonal artifacts. Computer-based elicitation techniques have proved capable of supplying these skills, essentially acting as expert systems for grid elicitation. PLANET is an integrated suite of elicitation and analysis programs designed for widespread application by users with widely varying backgrounds. This paper presents our experience in the development, dissemination, and application of PLANET. The subprograms described are: PEGASUS, which elicits from a person the constructs he or she is using; SOCIOGRIDS, which analyzes the relationships between members of a group through their differing construct systems and determines consensual constructs; MINUS, which compares two grids; CORE, which extracts the invariant core constructs; FOCUS, which hierarchically clusters constructs and elements; PRINGRID, which gives a principal components analysis; and ENTAIL, which analyzes the logical dependencies between constructs and can be used to develop rules for an expert system shell that acts as a cognitive model of the person.

During the past 20 years, the digital computer has come to assume a role of increasing importance in psychology and in the application of

psychological techniques. Also during this period, the nature of available computer resources has itself changed dramatically. In the early 1960s we submitted massive computations of analyses of variance and factor analyses to some remote computing center that punched our data onto cards and sent back the results, often days later. By the early 1970s we had our own minicomputers in our laboratories, which provided a hands-on system for direct experimental control, but were so expensive they were available only as a resource shared by a group of researchers on a booking basis. The central computers also changed during that period to allow some form of interactive time sharing whereby we could enter our own data on a local terminal and receive the analysis back reasonably rapidly on the same terminal. In the last decade, a further development has taken place in that so-called personal computers have become available, offering us great power and a wide range of facilities at such a low cost that an individual can have one or more such machines dedicated to his or her work.

This chapter is concerned with a set of repertory grid programs implemented as an integrated system on a low-cost microcomputer. This was done in such a way that all aspects of the operation and many aspects of the application of the system are self-evident to users who are issued with only a program disk and a simple manual. These users are widespread geographically, with a wide range of experience of computer systems, test procedures, analytic techniques, and information presenta-tion. It is also a requirement of some potential users that they be able to reprogram the interactive dialogue with their clients to reflect the particular purpose for which a grid is being elicited, for example personnel selection, career guidance, or industrial training. Some users also wish to translate the dialogue from English into their native language. The chapter also describes the techniques adopted to allow these modifications without requiring programming knowledge on the part of the users and without undermining the integrity of the program suite.

COMPUTER-BASED REPERTORY GRID ELICITATION AND ANALYSIS

The repertory grid and its applications are described in Chapters 1 and 2 of this book. In computer terms, the repertory grid is a simple data structure of entities (the elements) and attributes (the constructs) that is complete when each entity has been assigned a value for each attribute

(the ratings). In these terms a grid can be seen as part of an entity–attribute database (Chen 1980). In addition, with each grid it is useful to maintain explanatory and classificatory data, such as a list of subjects from whom it was elicited, and the time and purpose of elicitation.

REPERTORY GRID ELICITATION

In the past the elicitation of repertory grids has been a task requiring a skilled psychologist who can draw out a person's constructs while not imposing his or her own ideas or personality. This has involved the development of a variety of techniques and methodologies for grid elicitation (Kelly 1955; Fransella and Bannister 1977). The advent of powerful, low-cost personal computers has made it possible to program some of the elicitation techniques in a form in which they can be applied automatically. It is also possible to codify some of the techniques in a form in which they can be administered through a conversational interactive computer program. PLANET contains two such programs: PEGASUS, which elicits a single grid from a person and feeds back comments and suggestions during the elicitation; and ARGUS, which elicits a set of grids from a person using a number of different perspectives.

PEGASUS is a set of programs, each of which elicits a repertory grid (details of the variants are given later). The most 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. 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 with 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; to redefine the purpose for eliciting the grid. A flow chart of control around these options is shown in Figure 3.1.

ARGUS elicits a set of grids simultaneously from one person holding several roles or points of view. First, the elements, which are also the roles or points of view, are elicited, followed by three constructs from fixed triads. These construct labels are then used for a new set of ratings

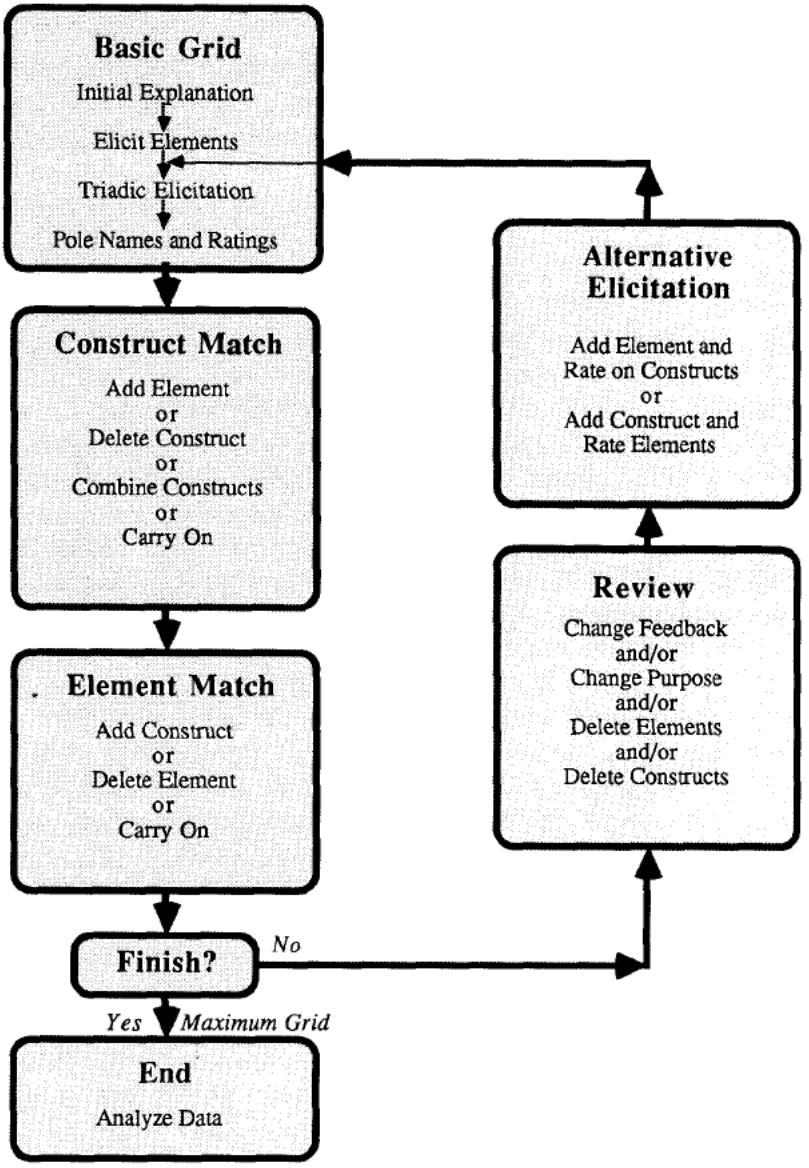

Fig. 3.1.
Flow of control in PEGASUS elicitation.

to be entered for each perspective in turn, and at each stage another construct that is felt to be important for that role is added. Eventually, each perspective has a grid using construct names from all the perspectives, but with unique rating values. The final set of grids, all with the same element and construct labels, can be processed with the structure analysis programs SOCIOGRIDS, CORE, or MINUS (described below).

REPERTORY GRID ANALYSIS

Repertory grids in themselves encode information about a person's way of looking at the world. This information can be used in its own right for some purposes, since it is an aid to remembering the basis for decisions and actions. It can also be analyzed in a variety of ways to bring out possible underlying structures, or construct systems, in a person's world view and its relationship to those of others. There are a number of forms of analysis that are widely used for different purposes, and PLANET offers all the commonly used techniques plus the developments of recent years. The form of analysis that should be used in a particular case is partly a matter of personal preference and partly a matter of purpose. Comparisons have been made in the literature of different analyses with the same data (Shaw 1981).

It is convenient to divide analysis techniques into three groups: those concerned with the analysis of a single grid; those concerned with a pair of grids; and those concerned with a group of grids.

ANALYSIS OF A SINGLE GRID

For any given construct we may regard the numbers in the grid as a vector of values, giving the assignment of each element in turn to one or the other of the poles of the construct. From this point of view each construct is represented as a point in a multidimensional space whose dimension is the number of elements involved. A natural relation to examine between constructs is then the distance between them in this space. Two constructs that are zero distance apart are such that all elements are construed in the same way in relation to them, and hence we might infer that they are being used in the same way; in some sense they are equivalent constructs. For constructs that are not equivalent, we may analyze the entire constellation in space to determine a set of axes such that the projection of each construct onto the first axis accounts for most

of the distance between them, the projection on the second axis accounts for most of the remaining distance, and so on. This is a principal components analysis of the construct space (Slater 1976, 1977), and it is related to the factor analysis of semantic space used in the study of semantic differentials (Osgood, Suci, and Tannenbaum 1957). We may also group constructs that are close together in space, using some form of cluster analysis (Shaw 1980).

The FOCUS algorithm (Shaw 1980) is a distance-based hierarchical cluster analysis technique that sorts the constructs into a linear order, such that constructs closest together in space are also closest together in the order. It has the advantage in presentation that the sorting is used only to re-present the original grid reorganized according to the proximity of constructs and elements. It is left to the user to construe the result in terms of his or her own personal meaning and to confirm this directly in terms of the original data. Element and construct matching scores are computed by subtracting the rating values in each position of the two elements or constructs, summing them, then mapping the result into 0 percent to 100 percent similarity for the elements, and 100 percent to –100 percent similarity for the constructs. Two matrices are thus formed, one for elements and one for constructs. Clusters are computed by selecting the highest numbers form these matrices (that is, the most similar in terms of the ratings given) and continuing this process for the next cluster, until all elements and constructs are incorporated. The tree diagrams show these clusters and are imposed on the re-sorted original grid data.

The PRINGRID algorithm is another distance-based cluster analysis using standard principal component analysis techniques and giving the same results as Slater's INGRID (Slater 1976, 1977).

The alternative to distance-based methods of grid analysis is a logical analysis, looking at constructs as predicates applying to elements. The ENTAIL algorithm (Gaines and Shaw 1981) derives asymmetric implications between the construct poles so that one can infer how a new element might be placed on one construct, given how it is placed on others. This is discussed in more detail with an example given in a later section.

ANALYSIS OF A PAIR OF GRIDS

It is desirable to compare several repertory grids. Shaw (1980) discusses exchange grids for the measurement of understanding and agreement between either two people or one person on two occasions.

For example, two people may each elicit a grid in an area of common knowledge or experience. Each may choose his or her 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 he or she uses those constructs on those elements and the other as he or 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.

There are three ways in which PLANET can be used to compare two grids. The first is to concatenate those grids that have the same elements, and then to use the algorithms of FOCUS, ENTAIL, or PRINGRID as if they were one large grid. One can then explore the interaction of ideas by examining mixed clusters of constructs from the two grids.

The MINUS algorithm requires the two grids to have the same element and construct names, and compares the two to indicate where changes in construing occur. This is done by superimposing one grid on the other and subtracting the equivalent rating values. The parts of the grids that are the same are then clearly visible, as is the extent of the variation elsewhere. The result shows where there was agreement indicated by areas of blanks, and grades of difference indicated by increasing numbers.

Again starting with two grids having the same element and construct names, the CORE algorithm examines the most changed element and construct alternately, identifies it, and allows it to be deleted from the grids. In this way the core elements and constructs that show agreement or understanding in the two grids are identified and recorded. This method has been used to chart change over time, and to evaluate courses by examining changes in the participant rather than using the traditional method of examining behavioral objectives.

ANALYSIS OF A GROUP OF GRIDS

The SOCIOGRIDS algorithms analyze a set of repertory grids elicited from a group that shares the same elements. SOCIOGRIDS compares every pair of grids using the FOCUS algorithm and produces a set of *socionets* which indicates the links of similar construing within the group. A *mode grid* is also produced that shows the constructs which are readily understood by the majority of the group, and this is filed for future processing on one of the structure analysis programs. Every

construct that has been used is listed in order of how highly matched it is across all grids, with a description of where it originated.

This technique can be used for a wide variety of purposes concerned with group structures, cultures, and relationships. It has been used for investigating the relative positions of members of a small group, and the content of the shared terms and values. It has been used to investigate management hierarchies defined operationally through the interrelationships of construct systems, and in conjunction with the Delphi technique to promote common values and understanding in a group.

KEY FEATURES OF THE PLANET SYSTEM

PLANET runs as a system of programs under the DOS 3.3 operating system on the Apple II family of computers. It requires two disk drives (referred to by the normal convention as drive 1 and drive 2) with the controller installed in slot 6. It also requires a printer usually available through slot 1 on the Apple. This is a standard operating configuration for the Apple II and //e.

```
****************************************
* PLANET  PERSONAL LEARNING, ANALYSIS, *
* NEGOTIATION & ELICITATION TECHNIQUES *
*           COPYRIGHT 1982             *
*   CENTRE FOR MAN-COMPUTER STUDIES    *
****************************************

 <P> PEGASUS    - ELICITATION

 <A> ARGUS      - MULTIPLE ELICITATION

 <S> STRUCTURE  - ANALYSIS TECHNIQUES

 <D> DATA       - DATA FILES UTILITY

 <O> OUTPUT     - OUTPUT DEVICE UTILITY

WHICH DO YOU WANT?
```

Fig. 3.2.
PLANET main menu.

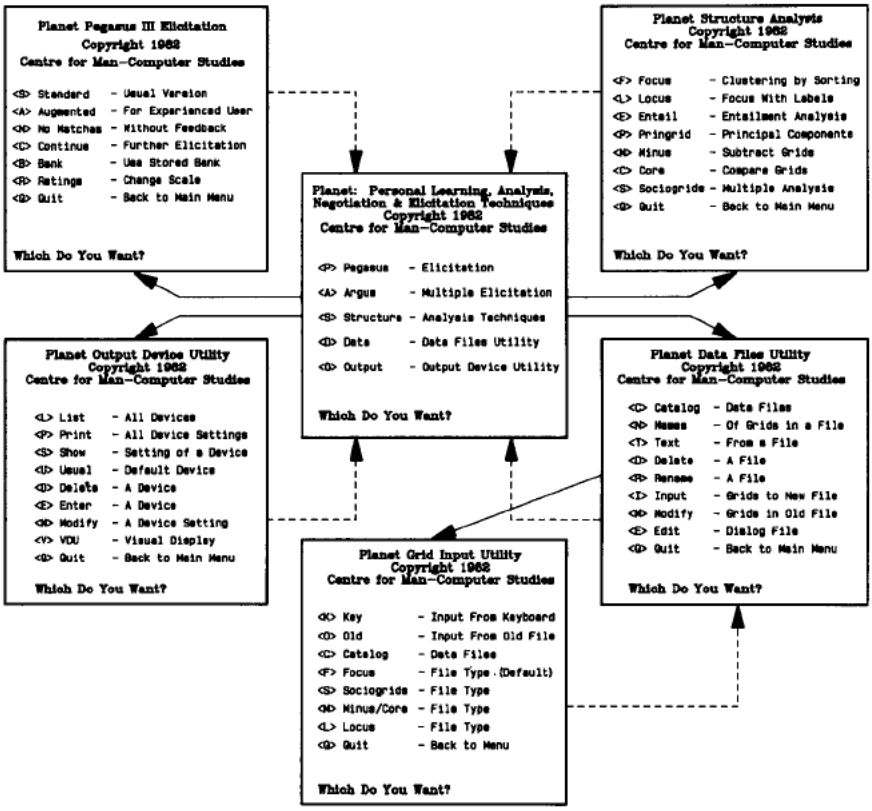

Fig. 3.3.
PLANET overall menu system.

To run PLANET, two disks are needed: the PLANET program disk and a data disk. The data disk can be any disk formatted for operation under DOS 3.3. PLANET is designed to check for possible errors in operation that may cause loss of valuable data, and checks that the data disk is valid before it commences any operations.

The main menu is both the starting and return point for all the PLANET programs. It has a PLANET logo as a heading followed by a list of the options, shown in Figure 3.2.

The reply to the question is simply the single character that appears in the angle brackets on the left-hand side of the chosen option. PLANET will fill in the rest and then call the program for the option requested.

When PLANET program options are complete they generally return to the main menu. If an error occurs in a program (such as a disk write being impossible because the disk is full), this is reported and a return is made to the main menu.

Figure 3.3 shows the overall PLANET system through printouts of the menus as they appear on the screen, together with the flow of control between them. Individual programs and utilities are described in the next two sections.

```
        PLANET PEGASUS III ELICITATION
              COPYRIGHT 1982
        CENTRE FOR MAN-COMPUTER STUDIES

<S> STANDARD    - USUAL VERSION

<A> AUGMENTED   - FOR EXPERIENCED USER

<N> NO MATCHES  - WITHOUT FEEDBACK

<C> CONTINUE    - FURTHER ELICITATION

<B> BANK        - USE STORED BANK

<R> RATINGS     - CHANGE SCALE

<Q> QUIT        - BACK TO MAIN MENU

WHICH DO YOU WANT?
```

Fig. 3.4.
PEGASUS menu.

GRID ELICITATION PROGRAMS

PEGASUS is an interactive program for eliciting a repertory grid that may have up to 15 elements and 15 constructs. Various versions that can be used are shown in the PEGASUS menu in Figure 3.4 and described in detail below. The menu in Figure 3.4 also shows the option to change the rating scale from the usual five points to anything from two points to nine points.

In the STANDARD version, six elements are chosen initially by the user, with special attention to the purpose for eliciting the grid. The first four constructs are elicited using fixed triads, and thereafter random or chosen triads are offered. Real-time processing gives continuous feedback about highly matched constructs and elements.

Options offered to the user in PEGASUS include:

- adding an element to split highly matched constructs
- replacing two highly matched constructs by one
- adding a construct to split highly matched elements
- deleting one or more element
- deleting one or more construct
- adding a construct without using a triad
- adding an element
- changing the level of feedback commentary
- redefining the purpose for eliciting the grid

The AUGMENTED version of PEGASUS is identical to the STANDARD version, except that extra elements may be added initially. It is intended for the fluent PEGASUS user.

The NO MATCHES version of PEGASUS gives a conventional triadic grid elicitation without any feedback, but with the opportunity to add and delete elements and constructs where the user wishes.

The CONTINUE version of PEGASUS allows the user to continue an elicitation started earlier. Otherwise, CONTINUE is identical to the STANDARD version.

The BANK version of PEGASUS provides an *expert* grid, which the user does not see at first but against which the elicited constructs are matched. Feedback is given not only on how the user's constructs match each other, but also on how they relate to the expert constructs. The final grid is put on file, and can be processed on FOCUS (or another of the

STRUCTURE programs) to show how the two sets of constructs are interrelated. The expert grid should be entered initially using one of the previous versions of PEGASUS.

The RATINGS option allows the rating scale being used in any version of PEGASUS to be changed. It is set by default to be 1 through 5, but the upper limit may be changed to be anywhere between 2 and 9. For those who frequently use a rating scale other than 1 to 5, a facility has been added to the PEGASUS menu where typing the value (i.e., 2 through 9) rather than R for RATINGS sets it up immediately. The rating scale range is now also shown in the menu.

ARGUS elicits a set of grids simultaneously from one person who is instructed to work from several roles or perspectives. First, the elements (significant others) are elicited, then three constructs are elicited from fixed triads in the usual way. These construct labels are then used for a new set of ratings, to be filled in from each perspective in turn. At each stage another construct that is felt to be important for that perspective is added. Finally the set of grids, all with the same element and construct labels but with different ratings, are filed for future processing on FOCUS, CORE, or SOCIOGRIDS.

STRUCTURE PROGRAMS

The STRUCTURE menu gives access to all the grid analysis programs available in PLANET. All of these programs analyze a grid (or grids) from a file on the data disk and produce output on the printer. The STRUCTURE menu in Figure 3.5 shows all the analysis programs available.

FOCUS gives the two-way hierarchical cluster analysis previously described. It will analyze grids with up to 54 elements and 54 constructs. Whether or not it has space to print them out depends on the printer being used. FOCUS checks the width specified for the printer and will print out the focused grid either in one piece complete with construct tree, or in two pieces with the construct tree separate, to be attached to the right-hand side of the focused grid.

LOCUS is a version of FOCUS that prints out the element and construct labels. It is especially useful with a PEGASUS grid or for displaying grids. It can be used for either the raw or the clustered version of the data, and will analyze up to 30 elements and 30 constructs.

ENTAIL gives the logical entailment analysis previously described. It will analyze up to 32 elements and 32 constructs. The implicational

```
PLANET STRUCTURE ANALYSIS
COPYRIGHT 1982
CENTRE FOR MAN-COMPUTER STUDIES

<F> FOCUS      - CLUSTERING BY SORTING
<L> LOCUS      - FOCUS WITH LABELS
<E> ENTAIL     - ENTAILMENT ANALYSIS
<P> PRINGRID   - PRINCIPAL COMPONENTS
<M> MINUS      - SUBTRACT GRIDS
<C> CORE       - COMPARE GRIDS
<S> SOCIOGRIDS - MULTIPLE ANALYSIS
<Q> QUIT       - BACK TO MAIN MENU

WHICH DO YOU WANT?
```

FIG. 3.5.
STRUCTURE menu.

relation of entailment is asymmetric, and hence ENTAIL gives a form of construct analysis that brings out order relationships and hierarchies if present.

PRINGRID is a principal components analysis program giving an analysis similar to Slater's INGRID (Slater 1976, 1977), presenting it both numerically and as graphic plots. It will analyze up to 30 elements and 30 constructs.

MINUS compares two grids as previously described. It highlights the differences between them by subtracting ratings in equivalent positions. A measure is given to indicate how similar the two grids are overall. It will analyze grids of up to 54 elements and 54 constructs.

CORE is an interactive program that compares two grids as previously described. The two grids of up to 54 elements and 54 constructs are processed by successively determining the element and construct that have changed the most between the two grids. The user is then offered the opportunity to delete the element or construct at each stage, finally leaving the core or unchanged part of the grids. The technique is essentially interactive, but the option is given to maintain a record of the matches, action taken, and final core grid on the printer.

SOCIOGRIDS analyzes a set of up to 20 repertory grids elicited from a group that shares the same elements as previously described. The number of elements can be up to 20, but must be constant across grids, as must the rating scale. The number of constructs can be up to 20 and

can differ from one grid to another. More details on these programs, their operation, and applications are given in Shaw (1980, 1981).

THE DATA UTILITY

DATA is a utility enabling an experimenter to examine, update, and maintain the experimental data and records of a program of personal construct elicitation using PLANET. The menu is shown in Figure 3.6.

CATALOG allows the files on the PLANET DATA disk to be catalogued by the user, and the output to be directed to the screen or the printer.

The NAMES of the grids in a particular file can be output with the details of the sizes of the corresponding grids onto the screen or the printer.

The TEXT of any data file on the PLANET DATA disk can be printed onto the screen or the printer.

DELETE allows the user to delete a data file.

RENAME allows the user to rename a data file.

The INPUT program is used to enter data for processing on any of the STRUCTURE programs. It will accept grids from within existing files, or from the keyboard, and forms a new file of grids. Typical applications of INPUT are entering grid data elicited by means other than

```
        PLANET DATA FILES UTILITY
            COPYRIGHT 1982
      CENTRE FOR MAN-COMPUTER STUDIES

<C> CATALOG    - DATA FILES
<N> NAMES      - OF GRIDS IN A FILE
<T> TEXT       - FROM A FILE
<D> DELETE     - A FILE
<R> RENAME     - A FILE
<I> INPUT      - GRIDS TO NEW FILE
<M> MODIFY     - GRIDS IN OLD FILE
<E> EDIT       - DIALOG FILE
<Q> QUIT       - BACK TO MAIN MENU

WHICH DO YOU WANT?
```

Fig. 3.6.
DATA menu.

the PEGASUS and ARGUS programs, and bringing together a number of grids elicited by PEGASUS into a single file for analysis by SOCIOGRIDS, MINUS, or CORE.

INPUT first asks the user for a file name for the new grid file. If a file of that name already exists, the user is given the option to overwrite it or to choose a new name. The menu then appears as shown in Figure 3.7.

Note that the QUIT command is particularly significant for INPUT, since it is only when this command is given that the new file being created is closed. Hence, any number and mix of grids may be created from data entered from the keyboard or from old files.

The KEY option allows grid data to be entered from the keyboard. Checks are made that the actual data entered corresponds in numbers of elements, constructs, and scale range to that specified.

The OLD option allows grid data to be entered from an existing file of grids. Again, checks are made to ensure that the data in the file satisfies any constraints specified.

The file type can be indicated initially, or at any time, and INPUT will check that grids entered previously and thereafter are compatible with the requirements of that file type. FOCUS allows any mix of grids in a file to be analyzed. SOCIOGRIDS requires that all grids have the same number of elements and rating scale range. MINUS and CORE require, in addition, that the number of constructs be the same. LOCUS operates on

```
        PLANET GRID INPUT UTILITY
              COPYRIGHT 1982
      CENTRE FOR MAN-COMPUTER STUDIES

<K> KEY          - INPUT FROM KEYBOARD
<O> OLD          - INPUT FROM OLD FILE
<C> CATALOG      - DATA FILES
<F> FOCUS        - FILE TYPE (DEFAULT)
<S> SOCIOGRIDS   - FILE TYPE
<M> MINUS/CORE   - FILE TYPE
<L> LOCUS        - FILE TYPE
<Q> QUIT         - BACK TO MENU

WHICH DO YOU WANT?
```

Fig. 3.7.
INPUT menu.

a file containing only one grid, together with the construct labels and element names, and its file type is unique in incorporating this extra data; it is the file type generated by PEGASUS. Setting the file type enables INPUT to perform checks to validate the data being entered. The initial file type set by default is FOCUS.

The MODIFY program can be used to edit data already on file. It steps through the grids in the file one at a time, giving options to add or delete elements or constructs, change particular values in the grid, or change the grid name.

EDIT copies the PEGASUS dialog file onto the PLANET DATA disk, so that it can be edited or rewritten (see section on PLANET PEGASUS Dialogue below).

PRINTER CHARACTERISTICS

Whereas the machine configuration, operating system, and language on the Apple II are more highly standardized than that of most microcomputers, the printers used vary considerably. Since one of the key features of the analysis programs in PLANET is meaningful layout of the clustered data, exact printer control is very important. The OUTPUT utility controls a general interpreter that handles all printer set-up commands and does this uniformly throughout the system. The interpreter is driven by a printer device file set up with the characteristics of a wide range of printers. The user can modify these using OUTPUT, adding additional ones as required, and changing the default printer characteristics being used. The interpreter allows for all the many variations of control characters to be sent to printers, status words to be modified in memory, slot changes to be made, and so on, and allows messages to be put out on the screen as to which printer is being used and when. The OUTPUT utility menu is shown in Figure 3.8.

LIST lists the devices available by name on the screen or printer.
PRINT prints the contents of the device file on the screen or printer.
SHOW shows the setting of a specified device.
USUAL sets up the default device to be used as that specified.
DELETE deletes a device from the device file.
ENTER enters a new device into the device file.
MODIFY modifies the characteristics of a specified device.

```
PLANET OUTPUT DEVICE UTILITY
        COPYRIGHT 1982
CENTRE FOR MAN-COMPUTER STUDIES


<L> LIST        - ALL DEVICES
<P> PRINT       - ALL DEVICE SETTINGS
<S> SHOW        - SETTING OF A DEVICE
<U> USUAL       - DEFAULT DEVICE
<D> DELETE      - A DEVICE
<E> ENTER       - A DEVICE
<M> MODIFY      - A DEVICE SETTING
<V> VDU         - VISUAL DISPLAY
<Q> QUIT        - BACK TO MAIN MENU


WHICH DO YOU WANT?
```

Fig. 3.8.
OUTPUT menu.

PLANET PEGASUS DIALOGUE

One of the new features of PEGASUS is that the actual dialogue with the user may be readily changed for different applications or national languages.

For example, the standard dialogue is intended to cover as many applications as possible and refers to general elements. It also has an explanation in terms of repertory grids. In a specific application such as market research for a particular class of products, psychiatric studies of particular problems, or personnel studies of relations with subordinates, it is desirable to have a dialogue reflecting the terminology usual in that application. Elements may become detergents, relatives, or staff; and the explanatory text may be revised to explain the particular activity being undertaken.

Clearly similar considerations apply to the use of PEGASUS for repertory grid elicitation in a language other than English. One important aspect of repertory grids is that they are language independent, with the entire vocabulary of the data coming from the person from whom the grid is elicited. However, in computer elicitation this also requires that the explanation and feedback be available in the language being used. PEGASUS now makes this possible.

The standard PEGASUS text file is issued on the program disk and is a DOS 3.3 text file (Type T). This file consists of blocks of free text on one or more lines separated by a line commencing with $$. Any text after the $$ is not used or stored by PEGASUS; it normally contains a number for the dialogue and possibly a comment on its use. An example is given in Figure 3.9.

The quotation marks show the whole of the text that is output, including extra new lines and spaces at the end of a line. A blank line causes a carriage return and a blank line to be output. There must be quotation marks if a , or " or : is included in the line, otherwise they are optional.

If a section of text needs to be omitted, the actual text lines must be deleted but not the $$ line following the text. For example, to delete the text in Section 8, it would become that shown in Figure 3.10.

If a line contains just # then a substitution occurs in the text either from a previous input by the user or calculated from the stage of the elicitation. An example is given in Figure 3.11, where the number of stored constructs is substituted in place of the #.

```
"$$15"
""
"Do you want to change the pole names"
"$$16 CHECK AFTER ELICITATION"
"Your element name is too long. Please choose a shorter one"
""
"$$17 CHECK AFTER ELEMENT ENTRY"
```

Fig. 3.9.
PEGASUS dialogue file showing the separators.

```
......
"These elements will then be used to elicit constructs."
""
"##"
"$$7"
"$$8"
"You have ......
```

Fig. 3.10.
PEGASUS dialogue file showing the effect of deletions.

If these parameters are required to be output in a different order or used more than once in the output sequence, they can be numbered to indicate this. An example of this is shown in Figure 3.12.

If the line that should contain a # is left empty, then the next line of text is omitted from the dialogue.

Where a line of text starts with a $ this has the effect of clearing the screen before outputting the following text (for example, see Figure 3.13).

If the $ occurs at the end of a line of text, it causes a carriage return (for example, in Figure 3.14).

```
"This grid already has "
"#"
"constructs$"
```

Fig. 3.11.
PEGASUS dialogue file showing the substitution of a number at runtime.

```
"Think of another element which is either"
"#4"
"and"
"#3"
"or"
"#1"
"and"
"#2"
```

Fig. 3.12.
PEGASUS dialogue file showing the output of parameters in a different order.

```
"$Name six elements."
""
.....
```

Fig. 3.13.
PEGASUS dialogue file showing how to clear the screen before output of text.

```
"Your pole name is too long please use a shorter one.$"
"$$59"
```

Fig. 3.14.
PEGASUS dialogue file showing how to output a carriage return.

The text shown in Figure 3.15 is part of an interaction with the general PEGASUS dialogue as it appears on the screen. This is generated by the part of the dialogue file shown in Figure 3.16. The next piece of text in Figure 3.17 shows the interaction in a specific market research context. This was generated from a new dialogue file written specifically for this market research, part of which is shown in Figure 3.18.

```
The two elements Margaret and Jane are
matched at the 86 percent level.

This means that so far you have not
distinguished between Margaret and Jane
Do you want to split these?yes

Think of a construct which separates
these two elements, with Margaret on
the left pole and Jane on the right
pole.

Left pole rated 1?close to me

Right pole rated 5?remote
```

Fig. 3.15.
An interaction with the standard PEGASUS dialogue.

Note how the matching level parameter "86" is not used in this case. Note also how the word-processing facilities of the dialogue interpreter fit the text to the size of the screen regardless of the parameters passed, such as element names.

The dialogue file is in upper and lower case. On an Apple II+ it will still appear in upper case. On an Apple IIe or IIc the PEGASUS menu gives an option of UPPER to switch it to upper case, or LOWER to switch back. If the 80-column card is present it will be used, and the

```
$$49
"$The two elements"
#
"and"
#
"are matched at the"
#
"percent level."
""
"This means that so far you"
"have not distinguished between"
#
"and"
#
"Do you want to split these"
$$50
""
"Think of a construct which separates these"
"two elements, with"
#
"on the left pole and"
#
"on the right pole.$"
$$51
""
"Left pole rated"
#
$$52
""
"Right pole rated"
#
$$53
```

Fig. 3.16.
The PEGASUS dialogue file generating the dialogue in Figure 3.15.

```
The detergents DAZ and PERSIL
seem very similar to me.

None of the ways of thinking about them
that you have mentioned so far makes
DAZ or PERSIL very different for you.
Can you think of a difference?Y

Tell me the difference which puts DAZ on
the left end of a scale and PERSIL on
the right end

Left end rated 1?I LIKE THE COLOR

Right end rated 5?NOT SO MUCH
```

Fig. 3.17.
An interaction with a PEGASUS dialogue file specifically for market research.

option FORTY will appear in the PEGASUS menu to force 40-column display, or EIGHTY to switch back.

PEGASUS will also accept lower-case input for purpose, elements, and constructs. Other input in lower case is converted to upper case, so that both "Y" and "y" or "YES" and "yes" are acceptable forms of affirmative reply. Often in PEGASUS the user is asked for a yes/no response and then for a number. Users sometimes type the number directly, omitting the "Y" for yes. PEGASUS allows this as an option where it makes sense.

PLANET is available also for the IBM PC, with only minor differences from the Apple version.

APPLICATIONS OF PLANET

The PLANET system, or parts of it, have been used in many studies. Reported in Shaw (1980) are a study on inspection of knitwear that was conducted over a hierarchical cross section of the firm using FOCUS and SOCIOGRIDS; a study on staff appraisal with the management services department of a paints division of a large company using PEGASUS, SOCIOGRIDS, and CORE; a study of personal and family relationships of teenagers in a psychiatric adolescent unit using FOCUS and CORE; and a study of role perspectives in an individual, using ARGUS,

```
$$49
"$The detergents"
#
"and"
#
"seem very similar to me.
""
"None of the ways of thinking about them"
"that you have mentioned so far makes"
#
"or"
#
"very different for you."
"Can you think of a difference"
$$50
""
"Tell me the difference which puts"
#
"on the left end of a scale and"
#
"on the right end$"
$$51
""
"Left end rated"
#
$$52
""
"Right end rated"
#
$$53
```

Fig. 3.18.
The PEGASUS dialogue file generating the dialogue in Figure 3.17.

FOCUS, CORE, and SOCIOGRIDS. It has also been used in education to study the changing constructs of the postgraduate (Phillips 1981) and the training of teachers (Pope and Keen 1981), in clinical psychology to study patient interview data (Shepherd and Watson 1982), in management and training (Steward and Stewart 1981), and in the transfer of knowledge from an expert to an expert system (Shaw 1983; Boose 1984b, 1986).

BIAIT: AN APPLICATION OF PLANET TO EXPERT SYSTEMS

The initial success of expert system (ES) developments (Michie 1979; Gevarter 1983; Reitman 1984) and the development of a number of reasonably domain-independent software support systems for the encoding and application of knowledge (Hayes-Roth, Waterman, and Lenat 1983) has opened up the possibility of widespread use of ESs. In particular, the Japanese Fifth Generation Computing System (FGCS) development program (Moto-oka 1982; Gaines 1984) assumes this will happen and has targeted knowledge processing rather than information processing. However, as noted, *knowledge engineering* represents a major bottleneck impeding the application of ESs in new domains.

The problems of knowledge engineering have been stated clearly:

> Knowledge acquisition is a bottleneck in the construction of expert systems. The knowledge engineer's job is to act as a go-between to help an expert build a system. Since the knowledge engineer has far less knowledge of the domain than the expert, however, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem solving; thus the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts. (Hayes-Roth, Waterman, and Lenat 1983)

The problem of knowledge elicitation from a skilled person is well known in the literature of psychology. Bainbridge (1979) has reviewed the difficulties of verbal debriefing and notes that there is no necessary correlation between verbal reports and mental behavior, and that many psychologists feel strongly that verbal data are useless. However, the PLANET system has been used to bypass cognitive defenses, including those resulting from automatization of skilled behavior. An example of an

application of some of the major PLANET programs is in the evaluation of expertise in business record keeping. Sowa (1984) describes the record-keeping needs of a business that can be prescribed through the BIAIT analysis technique (Carlson 1979), based on seven features:

1. Does the supplier bill the customer, or does the customer pay cash?
2. Does the supplier deliver the product at some time in the future, or does the customer take the order with him?
3. Does the supplier keep a profile of the customer, or is every transaction a surprise?
4. Is the price negotiated or fixed?
5. Is the product rented or purchased?
6. Does the supplier keep track of the product after it is sold or not?
7. Is the product made to order or provided from stock?

Sowa notes that these features "seem almost obvious once they are presented, but finding the right features and categories may take months or years of analysis. The proper set of categories provides a structural framework that helps us organize the detailed facts and general principles of a system. Without them, nothing seems to fit, and the resulting system is far too complex or unwieldy."

The BIAIT features are examples of constructs applied to orders placed between a customer and supplier. They provide a useful test case with which to evaluate the personal construct elicitation methodology provided in PLANET. An empirical study is being carried out to determine whether PEGASUS can be used to elicit the BIAIT distinctions from persons with some accounting or business knowledge. Hence, the initial experiments have been based on taking a spectrum of persons with assumed expertise in business record keeping, from businesspeople and accounts to accounting students at various stages of their education, and using PEGASUS to elicit their constructs relating to business transactions. Again, in the interests of an initial open-ended approach, the construing has not been tightly constrained to relate to the record-keeping aspects of the transactions. It has been interesting to determine what distinctions are made that do not relate to record keeping, or relate to it in ways different from BIAIT.

PEGASUS IN ACTION

PEGASUS commences its interaction by asking for the purpose of the elicitation and then for a set of elements relevant to this purpose, as

shown in Figure 3.19. In the dialogue the computer output is shown in normal face and the human input in bold. One technique used by PEGASUS to elicit constructs is that of triadic comparison shown in Figure 3.20. Another technique is that of matching the ratings given to elements and asking for a construct that differentiates between two highly matched elements, as shown in Figure 3.21. A technique that allows PEGASUS to elicit additional elements is to match constructs across elements and ask for a new element that differentiates between two highly matched constructs, as shown in Figure 3.22.

Through a combination of triadic comparison, element matching, and construct matching, a PEGASUS interaction elicits from the user his or her system of constructs relating to the topic being studied. The PEGASUS flow chart shown in Figure 3.1 shows how the different parts link together. Figure 3.23 is a composite grid in which the top seven constructs are those from BIAIT applied to a standard set of elements related to business record keeping. The remaining nine constructs were elicited from a businessperson using PEGASUS.

ENTAIL IN ACTION

ENTAIL (Gaines and Shaw 1981, 1986; Shaw and Gaines 1982) gives a dependency analysis of a grid by listing logical entailments consistent with the data. The dependency analysis is also being extended to be interactive, so that it may lead to the elicitation of further elements if the system suggests dependencies with which the subject disagrees, or to the modification of ratings if the subject suggests dependencies that are not possible according to the data he or she has entered. The overall objective is to allow the subject to explore the dimensions of his/her conceptual structure, testing it against specific cases, and discussing its implications, until he/she is comfortable that the representation elicited reasonably represents his/her knowledge.

Figure 3.24 shows the ENTAIL analysis of the composite grid shown in Figure 3.23. The entailments are shown with two values, the first a truth value from 0 to 100 (e.g., L7 $\rightarrow$ L8, truth 75), and the second the uncertainty reduction generated by asserting this entailment, again in the range from 0 to 100 (e.g., L7 $\rightarrow$ L8 with truth 75 reduces the uncertainty in L8 by 57). The uncertainty reduction measures the information carried by the assertion and is used to ensure that trivial entailments consistent with the data are pruned (Gaines and Shaw 1985).

```
Type in your purpose for doing this grid?examining
business record keeping

You must choose a set of six elements keeping in mind
why you want to do this grid. They could be people,
events, pieces of music, pictures, books, or what you
want but whatever you choose they must be of the same
type and each must be well known to you. Try to
choose specific things. Now type each one after each
question mark.

Do not forget to press the RETURN key after each.

  Element 1?selling newspapers from a stand
  Element 2?selling antiques in a market
  Element 3?selling custom drapes
  Element 4?selling large appliances
  Element 5?hiring out a car to a customer
  Element 6?providing TV rental services

Here is a list of your elements.

  1  selling newspapers from a stand
  2  selling antiques in a market
  3  selling custom drapes
  4  selling large appliances
  5  hiring out a car to a customer
  6  providing TV rental services

Do you want to change an element?no

Can you think of another element that belongs with the
6 that you have got so far?y

  Element 7?selling car insurance
```

Fig. 3.19.
PEGASUS initial element selection.

```
Triad for elicitation of construct 1
  1 selling newspapers from a stand
  2 selling antiques in a market
  3 selling custom drapes

Can you choose two of this triad of elements which
are in some way alike and different from the other
one.
What is the number of the one which is different?1

Now I want you to think about what you have in mind
when you separate the pair from the other one. How
can you describe the two ends or poles of the scale
which discriminate selling antiques in a market and
selling custom drapes on the left pole from selling
newspapers from a stand on the right pole?
Just type one or two words for each pole to remind
you what you are thinking or feeling when you use
this construct.

  left pole rated 1 --?no. of different objects
  right pole rated 5 --?same object in quantity

Now assume that selling antiques in a market and
selling custom drapes are assigned the value 1 and
selling newspapers from a stand is assigned the value
5 according to how you feel about them please assign
to each of the other elements in turn a provisional
value from 1 to 5

   2 selling antiques in a market      1
   3 selling custom drapes             1
   1 selling newspapers from a stand   5
   4 selling large appliances         ?3
   5 hiring out a car to a customer   ?1
   6 providing TV rental services     ?2
   7 selling car insurance            ?4
   8 providing electricity for home   ?5
   9 providing dental services        ?2
  10 being agent for home purchase    ?3
```

Fig. 3.20.
PEGASUS triadic construct elicitation.

```
The two elements 1 selling newspapers from a stand
and 8 providing electricity for home are matched at
the 87 percent level. This means that so far you have
not distinguished between selling newspapers from a
stand and providing electricity for home.
Do you want to split these?Y

Think of a construct which separates these two
elements, with selling newspapers from a stand on the
left pole and providing electricity for home on the
right pole.

  left pole rated 1 --?supplied one at a time
  right pole rated 5 --?direct delivery

According to how you feel about them please assign to
each of the other elements in turn a provisional
value from 1 to 5

  1 selling newspapers from a stand  1
  8 providing electricity for home   5
  2 selling antiques in a market     ?1
  3 selling custom drapes            ?2
  4 selling large appliances         ?1
  5 hiring out a car to a customer   ?1
  6 providing TV rental services     ?4
  7 selling car insurance            ?3
  9 providing dental services        ?1
 10 being agent for home purchase    ?1
```

Fig. 3.21.
PEGASUS construct elicitation by breaking element match.

```
The two constructs you called
  5 direct delivery -- supplied one at a time
  9 'hands-on' delivery -- nonpersonal delivery
are matched at the 75 percent level

This means that most of the time you are saying
direct delivery you are also saying 'hands-on'
delivery and most of the time you are saying supplied
one at a time you are also saying nonpersonal
delivery. Think of another element which is either
direct delivery and nonpersonal delivery or 'hands-
on' delivery and supplied one at a time. If you
really cannot do this then just press RETURN after
the first question mark but please try.

Then you must give this element a rating value on
each construct in turn. After each question mark type
a value from 1 to 5

What is your element?
```

Fig. 3.22.
PEGASUS element elicitation by breaking construct match.

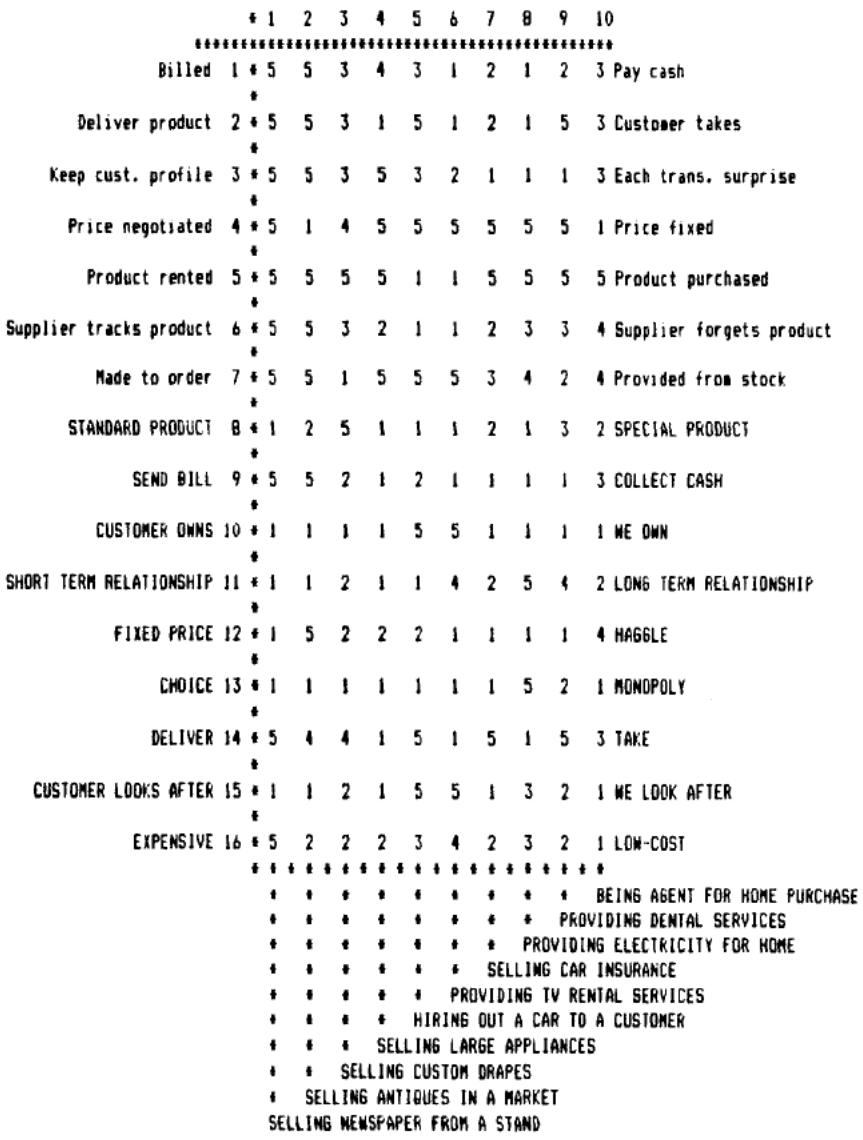

Fig. 3.23.
Composite repertory grid. BIAIT constructs at top; constructs elicited from businessperson by PEGASUS at bottom.

```
Entail   Truth  Information (Cutoff 30)

L 7 -> R 8    75  57 Made to order -> SPECIAL PRODUCT
R 3 -> R 1    75  53 Each trans. surprise -> Pay cash
R 9 -> R 1   100  52 COLLECT CASH -> Pay cash
L 1 =  L 3    75  51 Billed  = Keep cust. profile
R11 -> L 1   100  51 LONG TERM RELATIONSHIP -> Billed
R11 -> L 3   100  51 LONG TERM RELATIONSHIP -> Keep cust. profile
R 1 =  R 3    75  51 Pay cash  = Each trans. surprise
R15 -> L 6   100  50 WE LOOK AFTER -> Supplier tracks product
R 3 -> L11   100  49 Each trans. surprise -> SHORT TERM RELATIONSHIP
R 9 -> R 3   100  45 COLLECT CASH -> Each trans. surprise
L 1 -> R11    50  43 Billed  -> LONG TERM RELATIONSHIP
R 1 -> L11   100  43 Pay cash  -> SHORT TERM RELATIONSHIP
L14 -> L 2    75  41 DELIVER -> Deliver product
L 5 -> L 6   100  40 Product rented -> Supplier tracks product
L 6 -> L 5    50  40 Supplier tracks product -> Product rented
L 4 -> R12    75  40 Price negotiated -> HAGGLE
L 5 -> R10   100  40 Product rented -> WE OWN
L 5 -> R15   100  40 Product rented -> WE LOOK AFTER
L 6 -> R10    50  40 Supplier tracks product -> WE OWN
R10 -> L 5   100  40 WE OWN -> Product rented
R10 -> L 6   100  40 WE OWN -> Supplier tracks product
R12 -> L 4    75  40 HAGGLE -> Price negotiated
R15 -> L 5    50  40 WE LOOK AFTER -> Product rented
R10 -> R15   100  40 WE OWN -> WE LOOK AFTER
R15 -> R10    50  40 WE LOOK AFTER -> WE OWN
L 3 -> L 9   100  39 Keep cust. profile -> SEND BILL
R 6 -> R 9    50  39 Supplier forgets product -> COLLECT CASH
L 8 =  R 7    75  38 STANDARD PRODUCT = Provided from stock
R 7 =  L 8    75  38 Provided from stock = STANDARD PRODUCT
R 9 -> R 2   100  38 COLLECT CASH -> Customer takes
L 1 -> L 9   100  37 Billed  -> SEND BILL
L 2 -> L 9   100  36 Deliver product  -> SEND BILL
R 6 -> L15   100  36 Supplier forgets product -> CUSTOMER LOOKS AFTER
R12 -> R 9    75  35 HAGGLE -> COLLECT CASH
R13 -> R11   100  35 MONOPOLY -> LONG TERM RELATIONSHIP
```

Fig. 3.24.
Entailment analysis of repertory grid in Figure 3.23.

In this example, the ENTAIL analysis is being used to see which BIAIT constructs may be recovered from the businessperson. Usually entailment analysis is used to derive relations between constructs for use in expert system shells. The entailments derived by ENTAIL can be converted directly to rules that may be loaded into a standard expert system shell, as Boose (1984) has shown. The truth value of the rule codes into the truth value representation used by the shell. The rules may be checked by the expert for their content but their prior evaluation for uncertainty reduction makes it unlikely that spurious rules will be generated.

FOCUS IN ACTION

Figure 3.25 shows the FOCUS analysis of the grid given in Figure 3.23. The construct matches are shown by the tree drawn on the right side of the FOCUS grid. It can be seen from the high construct matches that C5 and C10 are matched at 100 percent, — that is, we own products that are rented, whereas the customer owns products that are purchased; we haggle over products where the price is negotiated, whereas we have a fixed price if the price is fixed. Two highly matched BIAIT constructs show up in cluster 19. Each transaction is a surprise when the customer pays cash, whereas a customer profile is kept when the customer is billed. We look after rented products that we own, whereas the customer looks after purchased products that he or she owns, in cluster 20; and standard products are provided from stock, whereas special products are made to order, in cluster 21.

It is easy to see the highly clustered elements, and the ones that stand out on their own in the tree diagram above the grid in Figure 3.25. The element clusters indicate that the record keeping for providing dental services and for selling car insurance are the most similar (cluster 11); and selling newspapers from a stand is similar in record keeping to selling antiques in a market. The most different one is E4, selling large appliances, which is not highly clustered to any of the other elements. Providing TV rental services is quite similar in its record keeping to hiring out a car, but these two are unlike any other examples of record keeping in this grid.

Most of the entailments in Figure 3.24 appear in the clusters matched over 50 percent, that is, up to cluster 27. We get some confirmation in these results that ENTAIL and FOCUS produce very similar results, but in a slightly different form. The ENTAIL results can be translated directly

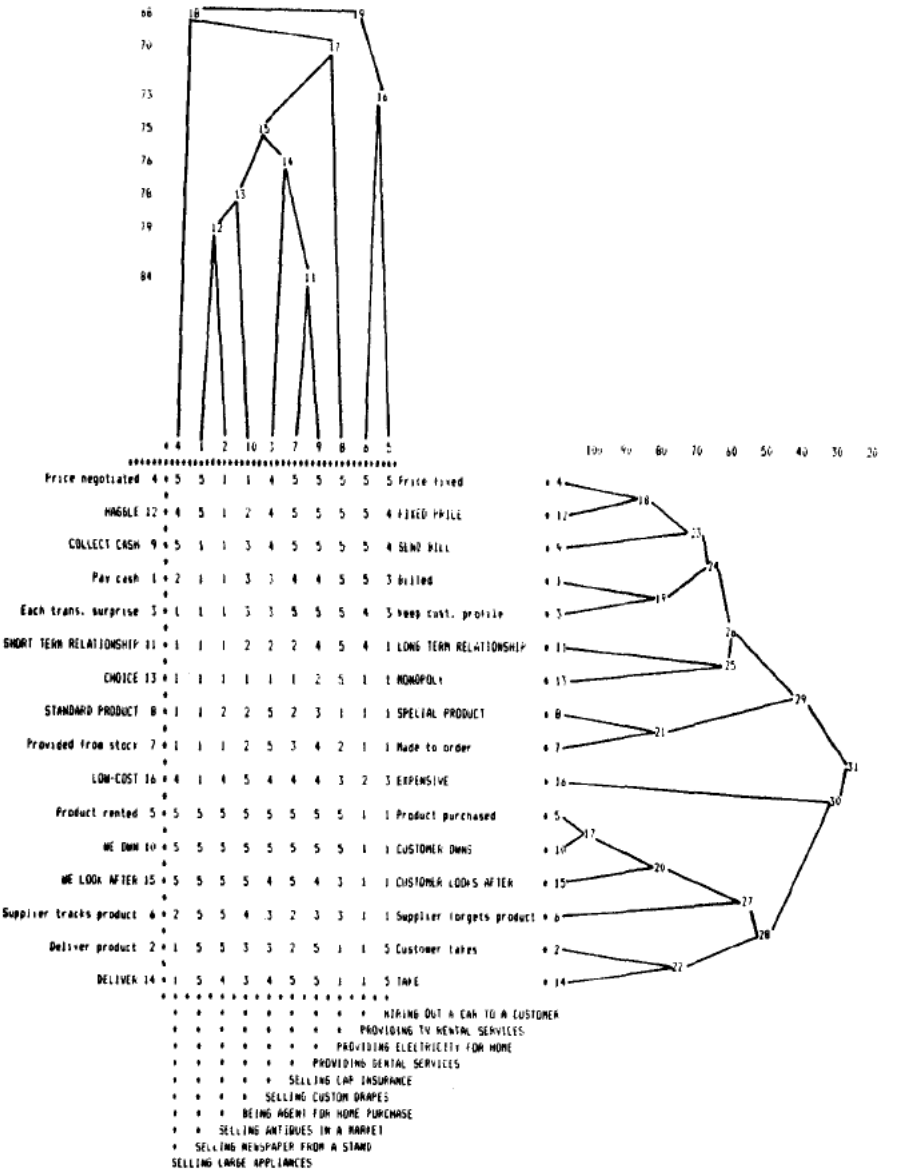

Fig. 3.25.
FOCUS analysis of repertory grid in Figure 3.23.

into production rules (Hayes-Roth, Waterman, and Lenat 1983). For example, the first one would become, "If a product is made to order, then it is a special product"; and the last one would become, "If there is a monopoly, then there is a long-term relationship with the customer." Most of these results seem intuitively obvious, even if one knows little of business record keeping.

SOCIOGRIDS IN ACTION

The construct elicitation described in the previous section is being carried out with a number of subjects, from those experienced in business record keeping to students with some knowledge of accounting. It is clearly possible to analyze for each subject whether the BIAIT constructs are among those elicited, how closely they are mimicked, and which ones are missing, and this analysis is being done. However, one can also see whether it is possible to reconstruct the ranking in expertise of the subjects through a SOCIOGRIDS analysis. For example, in an earlier study (Shaw 1980) of quality control for a garment company, the SOCIOGRIDS analysis reconstructed the management hierarchy of the firm from the construct systems of staff involved in construing garment faults.

The operation of SOCIOGRIDS is based on what Kelly (1955) terms the *commonality corollary*: To the extent that one person employs a construction of experience that subsumes that of another in a given domain, that person can understand the cognitive processes of the other person in that domain.

SOCIOGRIDS derives linkages between people (socionets) by comparing grids in pairs (A and B, say) to determine for each construct in grid A the best matching construct in grid B. The comparison is done by calculating the average over all constructs in grid A of the similarity of the best match for each construct in grid B. This measures to what extent the system of construction modeled in grid B subsumes that modeled in grid A. These subsumption averages are ordered and used to produce socionets showing the degree to which one person's construction system subsumes that of another. SOCIOGRIDS derives a consensual construction system (mode grid) by comparing each construct in turn with those in every grid and calculating the average over all grids of the best match for the construct in each grid. This measures to what extent a given construct is represented in the construction system of every person, and hence is consensual.

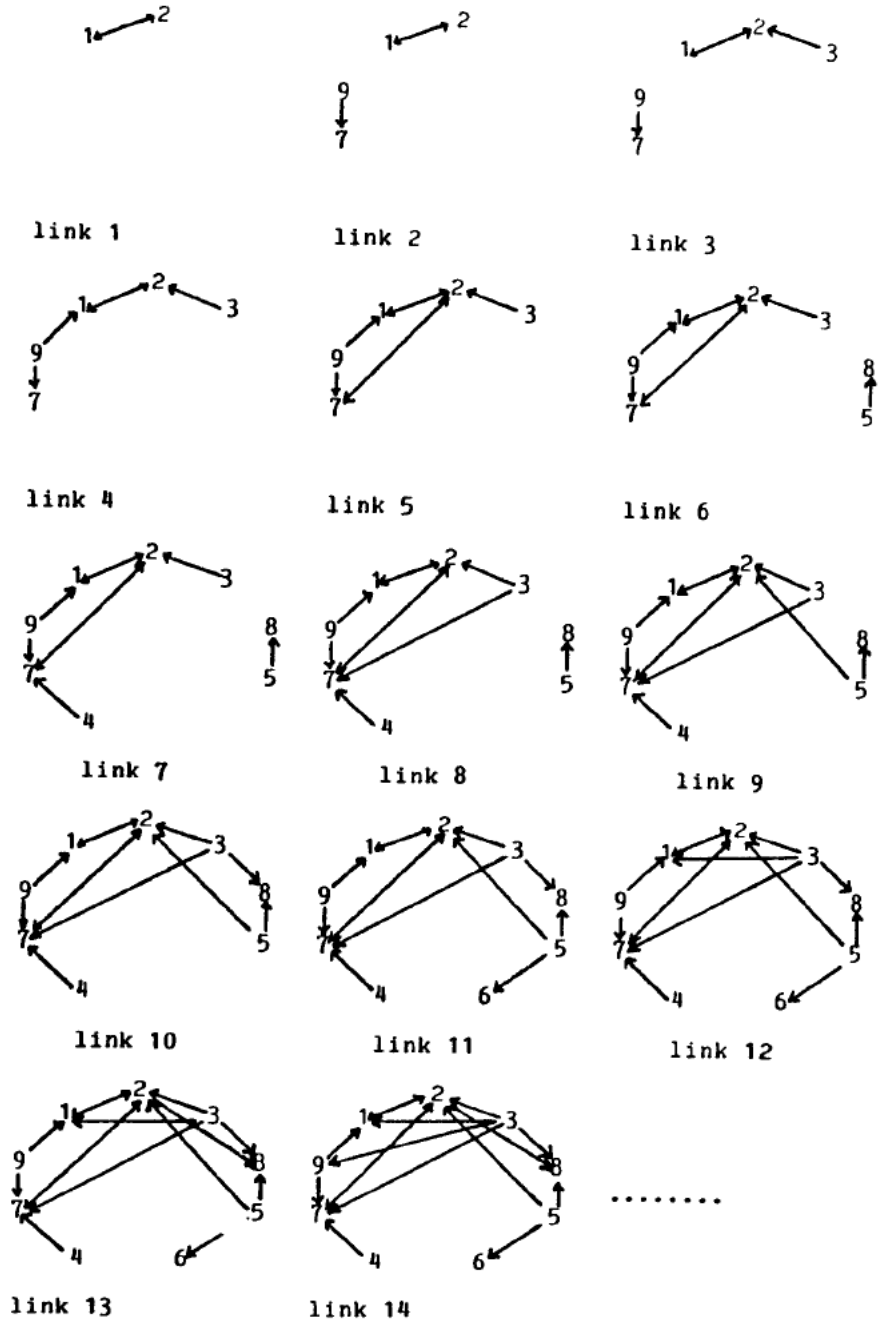

Fig. 3.26.
Series of socionets produced by SOCIOGRIDS from BIAIT validation experiment.

Link 1 in Figure 3.26 shows the most highly matched pair, which is BIAIT and a businessperson; link 2 shows another matched pair, which is an economics student and a visual arts student. As the links are formed these initial pairs are joined into one group at link 4. A new subgroup is formed at link 6, then gradually the other participants are brought in and joined into one group. The nodes forming the central part of the network are BIAIT, the two businesspeople, an economics student, and the visual arts student. This can be seen at link 14, which is shown more completely in Figure 3.27.

This is mainly what one would expect, with the businesspeople and the economics students closely related to BIAIT. The only disconcerting result is the position of the visual arts student, who is clearly a central part of the cluster. On inquiring into the background of this student, it was discovered that she had only recently become a student, having spent the previous 25 years as an administrative secretary in an insurance organization.

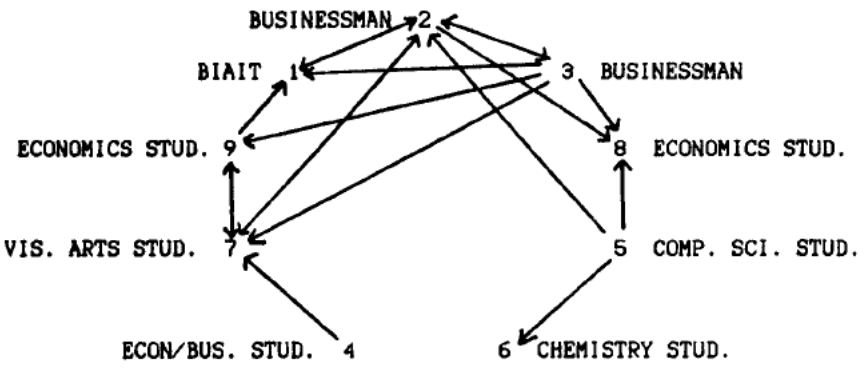

Fig. 3.27.
Socionet produced by SOCIOGRIDS from BIAIT validation experiment.

SUMMARY AND CONCLUSIONS

Repertory grids have proved to be powerful tools in a wide range of disciplines and applications. However, elicitation by people requires skills in interviewing techniques, personal construct psychology, and the avoidance of interpersonal artifacts. Computer-based elicitation techniques have proved capable of supplying these skills, essentially

acting as expert systems for grid elicitation. PLANET is a complete suite of elicitation and analysis programs designed for widespread application by users with widely varying backgrounds. This chapter has presented our experience in the development, dissemination, and application of PLANET.

PLANET has most recently been applied to knowledge engineering for expert systems. The conceptual framework involved is of widespread interest because in educational, clinical and management applications we are involved in eliciting personal knowledge representing individual "expertise," its internal structure, and its relations to the "expertise" of others. PLANET contains a number of subprograms to do this. PEGASUS elicits from a person the fundamental distinctions, or constructs, he/she is using in applying his/her expertise; SOCIOGRIDS analyzes the relationships between members of a group through their differing construct systems and determines consensual constructs; ENTAIL analyzes the logical dependencies between constructs and can be used to develop rules for an expert system shell.

Note that in non-expert-system applications, the rules operating in a shell become a cognitive model of the person from whom the constructs have been elicited and hence a means for anticipating his or her cognitive, decision-making, and behavioral processes. Such shells can give feedback to individuals on the dynamics of their personal thinking and can further enhance the reflective processes of constructive alternativism that Kelly saw at the heart of personal construct psychology.

We see PLANET as an expert system incorporating the best features of elicitation from skilled people and making these available to those who do not have the time, experience, or personality to develop the necessary skills. Those writing elicitation programs should note that the advantages described do not come automatically with the use of a computer. The design of the human–computer interface requires great care and takes rather more effort than the writing of the elicitation and analysis procedures in their own right. To make person–computer interaction as attractive as the best person–person interaction requires that much of the expertise of good personal interaction be encoded for the computer. Certainly most of the faults of poor personal interaction can be programmed for the computer unless great care is taken!

APPENDIX

The PLANET programs and some of the more difficult to obtain books are available from the Centre for Person–Computer Studies. For a detailed price list please write to:

Dr. M. L. G. Shaw
Centre for Person–Computer Studies
4 Edgepark Mews NW
Calgary, Alberta
Canada T3A 3J7

For those people who do not have access to or do not wish to use a computer, many of the techniques used in PLANET and others are explained, with examples, for use with paper and pencil in Shaw and McKnight (1981b).

Since this chapter was written, a new set of programs has been developed for the Apple Macintosh. This incorporates many of the PLANET programs and new exchange procedures in an integrated package using the Macintosh graphic interface. It will be available shortly from the above address.

ACKNOWLEDGMENT

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