Int. J. Man–Machine Studies (1982) **17**, 345–360

PLANET: some experience in creating an integrated system for repertory grid applications on a microcomputer

MILDRED L. G. SHAW

Department of Computer Science, York University, 4700 Keele Street, Downsview, Ontario, Canada M3J 1P3 and Department of Psychiatry, Guy's Hospital Medical School, London, U.K.

(*Received 18 March 1982, and in revised form 25 June 1982*)

Direct interaction between patients and computers has now been shown to be successful and acceptable in an experimental environment. Automated psychological testing has also been widely validated against experts administering the same test. However, there are problems in making available what are essentially computer-based research tools to a diverse community of users, geographically widespread, with a range of experience of computer systems, test procedures, analytic techniques and information presentation. This paper is concerned with the experience of transforming a set of repertory grid programs developed for research purposes on a large mainframe computer into an integrated system on a low-cost microcomputer. This had to be done in such a way that all aspects of the operation and many aspects of the application of the system would be self-evident to users issued with only a program disk and simple manual. It was also a requirement of some potential users that they be able to re-program the interactive dialogue with their clients to reflect the particular purpose for which a grid was being elicited: for example, personnel selection, career guidance or industrial training. Some users also wished to translate the dialogue from English into their native language. The paper also describes the techniques adopted to allow this without requiring programming knowledge on the part of the users and without undermining the integrity of the program suite.

Introduction

During the past 20 years the digital computer has come to assume a role of increasing importance in psychology and the application of psychological techniques. During this period also, 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 Centre that punched our data on to cards and sent back the results often days later. By the early-1970s we had our own minicomputers in our laboratories giving a *hands on* system for direct experimental control but being so expensive that they had to be available as a shared resource to 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 few years a further development has taken place in that so-called *personal* computers have become available offering us great power and a

0020–7373/82/070345 + 16 $03.00/0

wide range of facilities at such a low cost that an individual can have one or more such machines dedicated to his work.

These advances in computer technology have offered opportunities for the automation of psychological testing which have been widely taken (Elithorn & Telford, 1969; Gedye & Miller, 1969; Elwood, 1972*a*, *b*, *c*; Klinge & Rodziewicz, 1976; Thompson & Wilson, 1982; Volans & Levy, 1982). Direct interaction between patients and computers has now been shown to be successful and acceptable in an experimental environment (Card, Nicholson, Crean, Watkinson, Evans, Wilson & Russell, 1974; Lucas, 1977), and it has even proved possible for the computer to act as an effective therapist (Stodolsky, 1970). Automated psychological testing has also been widely validated against experts administering the same test (Ridgway, MacCulloch & Mills, 1982; Calvert & Waterfall, 1982; Watts, Baddeley & Williams, 1982; Acker, 1982). These advances in technology have also offered the opportunity for new approaches to the testing process dependent on having powerful computational facilities in operation during the test. In particular, it has become possible to obtain test results speedily, these being presented in a manner which aids both expert interpretation and self-interpretation. With the advent of the personal computer in particular, certain forms of test previously only capable of professional administration and interpretation are becoming widely available, and this raises questions of professionalism and ethics. However, it also raises more mundane technical and human factors problems of taking what have been essentially research tools in the laboratory and issuing them widely to a very diverse community of users.

This paper is concerned with the experience of transforming a set of repertory grid programs developed for research purposes on a large mainframe computer (Shaw, 1980) into an integrated system on a low-cost microcomputer. This had to be done in such a way that all aspects of the operation and many aspects of the application of the system would be self-evident to users who are geographically widespread, with a wide range of experience of computer systems, test procedures, analytic techniques and information presentation, issued with only a program disk and simple manual. It was also a requirement of some potential users that they be able to re-program the interactive dialogue with their clients to reflect the particular purpose for which a grid was being elicited: for example, personnel selection, career guidance or industrial training. Some users also wished to translate the dialogue from English into their native language. The paper also describes the techniques adopted to allow this without requiring programming knowledge on the part of the users and without undermining the integrity of the program suite.

Transporting the programs

The PLANET suite is based on a set of repertory grid elicitation and analysis programs developed to show that the *Personal Construct Psychology* of George Kelly (1955) could be operationalized through computer algorithms and given widespread practical application (Shaw, 1980). As the availability of these programs became known requests were received for their use from many individuals and institutions in some 10 different countries and wishing to use some 15 different computers. The programs were primarily available on a DECsystem-10 time-sharing mainframe computer and some users were able to access this over the telephone network in Britain. However, other users were concerned to have the programs available on their local machines and they were transferred to a range of computers including a number of mainframes, minicomputers and microcomputers.

In theory the transfer of programs to a range of machines should not have been difficult, since they were written in BASIC and used only the standard language constructs specified for Dartmouth College BASIC. However, the experience if transporting the programs highlighted many problems of differences between BASIC implementations. Some of these are widely known, such as the lack of standardization in file accessing facilities and commands. Others are more subtle but also widely known, such as the difference between implementations which execute a FOR/NEXT loop at least once and those which test the index before the first execution. Some defects in BASIC implementations were unexpected and caused major problems; in particular the string primitives not only differed in syntax and action but also in the space available, garbage created and effect of garbage collection. For construct elicitation programs involving large amounts of variable dialogue, the defects in string implementation on some machines were major restrictions. Add to these problems the lack of standardization, even lack of availability on some systems, of techniques for modularization of programs and program suites, such as chaining with parameter passing, and then the effort of transporting software becomes excessive.

Some will see all these problems as criticism of the language BASIC. However, similar problems arise in transporting software in dialects of FORTRAN, COBOL or PASCAL from one machine to another, particularly if extensive use is made of text files, interactive dialogue and string handling. BASIC is one of the few languages supporting variable-length text strings effectively and efficiently, at least in many of its implementations, and is still the only widely available language in which dialogue can be written simply. One surprising outcome of this experience in transferring the repertory grid programs between machines was to note that low-cost microcomputers often had far better BASIC implementations than either mainframes or minicomputers. One of the larger programs, PEGASUS, was transferred from the PDP10 to the Apple II and Commodore PET with little modification, but came up against severe space limitations on much larger minicomputers and mainframes.

However, all these problems of software portability are minor in relation to the problems of updating and supporting diverse users on different machines. Repertory grid elicitation and analysis techniques are rapidly developing (Shaw, 1981) and the dissemination of new programs cannot be achieved if it involves the effort that has been necessary in transporting the original limited suite. Hence, a project was started in 1981 to integrate the programs that now form the PLANET system and to issue them as a package that could be made widely available and supported with updates on a continuing basis. It was decided to start with just one microcomputer system, the Apple II, since this was widely available, the cost of acquisition was low, and the BASIC was completely adequate and supported by a range of utilities including a compiler. The machine also had the advantage over potential competitors that its operating system was relatively stable and the configurations available seemed to have become relatively standard. The only significant disadvantages were the upper-case only character set, 40-character wide screen and the comparatively low storage capacity disks.

The PLANET suite

Many of the programs in the PLANET suite have been described in detail elsewhere (Shaw, 1980). However, one of the first effects of integrating the programs and distributing them to a variety of users was to highlight the need for a manual to go with them which was not a research text, but rather a concise presentation of their use. The following sections are extracts from the user manual and demonstrate not only the nature of the programs but also the differing style of presentation needed for such a manual.

WHAT IS PLANET?

PLANET consists of a set of computer programs for the elicitation and analysis of repertory grid data from one or more people. It is based on the Personal Construct Psychology of George Kelly (1955), made operational through the personal computer. PLANET is a conversational tool for investigating the basis of your, and others', thinking with few constraints and presuppositions. It may be used in a wide variety of applications ranging from personal decision making through psychiatric help to the study of cultures. It may be used in management systems, for the evaluation of training courses and for staff appraisal. PLANET is designed to operate as an integrated set of programs accessed from menus and providing a number of database management utilities on the Apple II computer. It offers a wide range of alternative approaches to grid data analysis and presentation, together with forms of conversational computer interaction for both naive and professional users.

WHAT CAN YOU DO WITH PLANET?

Here are some of the things you can do with PLANET. You can elicit a conventional repertory grid or a grid with on-going immediate feedback of similar constructs or elements. You can start to elicit a new grid or continue one previously started. You can have your constructs matched against all your other constructs or against those of another individual or group on the same topic. You can elicit several grids from different perspectives at the same time. You can analyse individual grids by cluster analysis or by entailment analysis. You can analyse pairs of grids by examining the areas of identical ratings or by extracting the core constructs and elements in the grids. You can process grids from a group of people to extract those constructs common to the group and find the similarities in construing within the group.

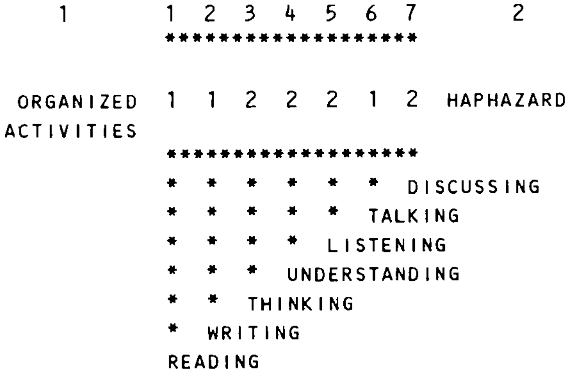
You can create your own PLANET DATA disk and create and store your own files of grids. You can list the files stored on your PLANET DATA disk on the screen or the printer, rename a file or delete a file. You can print the content of your data files on the screen or the printer, or just the numbers of elements and constructs, the rating scales and names of the grids in a file. You can edit your data files to process your grids in the ways you choose on any program, or copy the PEGASUS dialogue file from the PLANET program disk to your DATA disk for re-writing or editing. You can enter the special requirements for your printer or for several different printers. You can list these settings, modify or delete them. You can specify the screen display for a 40-column standard machine or an 80-column card in slot 3 or elsewhere.

WHAT IS A REPERTORY GRID?

A *repertory grid* is a two-way classification of data in which events are interlaced with abstractions in such a way as to expresss part of a person's system of cross-references between his *personal observations* or experience of the world (*elements*), and his *personal constructs* or classifications of that experience.

The *elements* are the things which are used to define the area of the topic. They may be physical objects, people, events, or abstract entities such as careers. Before choosing the set of elements, the user must think carefully about the area of the topic and relate the elements to his purpose. The elements should be of the same type and level of complexity, and span the topic as fully as possible. It is usual to start with between 6 and 12 elements.

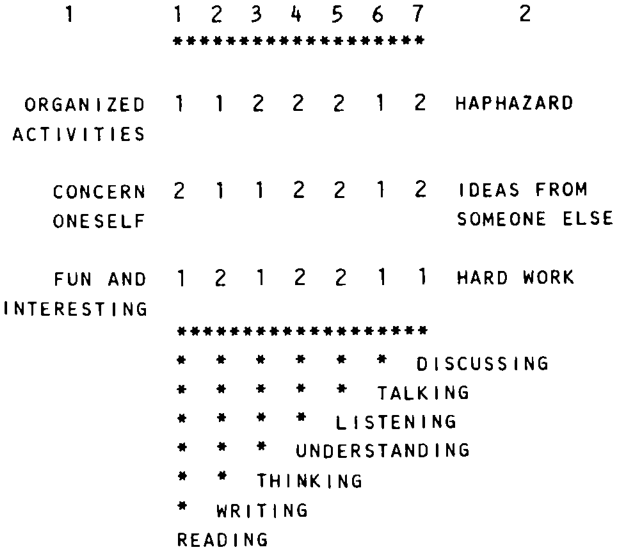
The *constructs* are the terms in which the elements are similar or different from each other. Each construct therefore has two poles, each of which has a meaning with respect to its opposite. The elements are presented in triads and a construct is elicited by saying in what way two are alike and why the other differs. For instance, thinking of the three activities of *reading*, *writing*, *thinking*, in what way are two alike and thereby different from the other one? We might first of all say that *reading* and *writing* are alike since they are *organized activities*, whereas *thinking* is *haphazard*. This is, then, the first construct with its two poles or opposite descriptions. This also shows the significance of the term *personal* in personal construct since it would not obviously be a publicly held description of these activities. Now all the elements in the set must be rated on this dimension as either 1 being *organized* or 2 being *haphazard*.



```
    1       1  2  3  4  5  6  7       2
            *******************

 ORGANIZED  1  1  2  2  2  1  2  HAPHAZARD
ACTIVITIES
            *******************
            *  *  *  *  *  *  DISCUSSING
            *  *  *  *  *  TALKING
            *  *  *  *  LISTENING
            *  *  *  UNDERSTANDING
            *  *  THINKING
            *  WRITING
            READING
```

FIG. 1. Activities as elements assigned to either pole of the construct *organized activities–haphazard activities.*

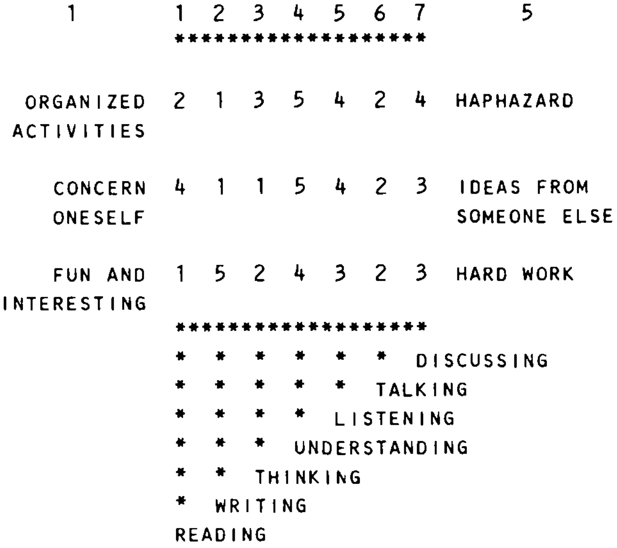
Then the second and subsequent constructs are elicited in exactly the same way choosing different triads each time. The third construct shown here is to illustrate that constructs can be factual, imaginary, pure emotion, or whatever is important to the user whose grid it is.



```
     1        1  2  3  4  5  6  7        2
              *******************

 ORGANIZED    1  1  2  2  2  1  2  HAPHAZARD
ACTIVITIES

   CONCERN    2  1  1  2  2  1  2  IDEAS FROM
   ONESELF                         SOMEONE ELSE

   FUN AND    1  2  1  2  2  1  1  HARD WORK
INTERESTING
              *******************
              *  *  *  *  *  *  DISCUSSING
              *  *  *  *  *  TALKING
              *  *  *  *  LISTENING
              *  *  *  UNDERSTANDING
              *  *  THINKING
              *  WRITING
              READING
```

FIG. 2. Additional constructs added to that of Fig. 1 to give a repertory grid with seven elements and three constructs.

In PLANET a scale allowing a range of distinctions beyond the pair 1 and 2 may be used as required. If a 1-5 scale were used then the above example might become:



```
     1        1  2  3  4  5  6  7        5
              *******************

 ORGANIZED    2  1  3  5  4  2  4  HAPHAZARD
ACTIVITIES

   CONCERN    4  1  1  5  4  2  3  IDEAS FROM
   ONESELF                         SOMEONE ELSE

   FUN AND    1  5  2  4  3  2  3  HARD WORK
INTERESTING
              *******************
              *  *  *  *  *  *  DISCUSSING
              *  *  *  *  *  TALKING
              *  *  *  *  LISTENING
              *  *  *  UNDERSTANDING
              *  *  THINKING
              *  WRITING
              READING
```

FIG. 3. A similar repertory grid to that of Fig. 2 with ratings on a 1-5 scale.

Thus, in this case, the first construct means that *writing* is considered the most *organized activity*, *thinking* is both *organized* and *haphazard*, and *understanding* is the most *haphazard activity*. Note how the use of a multi-point scale with an odd number of values allows for a central rating, in this case 3, which does not force the user to choose either pole.

REPERTORY GRID ELICITATION

In the past the elicitation of repertory grids has been a task requiring a skilled psychologist who can draw out from a person their constructs whilst not imposing their own ideas or personality. This has involved the development of a variety of techniques and methodologies for grid elicitation (Kelly, 1955; Fransella & Bannister, 1977). It is now possible to codify some of the techniques used by people in a form where 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.

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 analysed 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 new developments in recent years. What form of analysis 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), and if you are using repertory grids for the first time it is worth exploring the different techniques on your own data.

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 other of the poles of the construct. From this point of view each construct becomes represented as a point in a multi-dimensional 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 which 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 which are not equivalent we may analyse 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, and it is related to the *factor analysis* of semantic space used in the study of semantic differentials. We may also group constructs together that are close together in space using some form of *cluster analysis.*

The FOCUS algorithm is a distance-based hierarchical cluster analysis technique that sorts the constructs into a linear order such that constructs closest together in the 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 re-organized by the "neighbourness" of constructs and elements. It is left to the user to construe his own personal meaning into the result and confirm this directly in terms of the original 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 program 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.

ANALYSIS OF A PAIR OF GRIDS

There are often times when 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 two occasions. There are three ways in which PLANET can be used to compare two grids. The first is to concatenate the grids having the same elements and use 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.

MINUS requires the two grids to have the same element and construct names, and highlights the differences between them by subtracting ratings in equivalent positions. 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, CORE 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 which are unchanged in the two grids are identified and recorded.

ANALYSIS OF A GROUP OF GRIDS

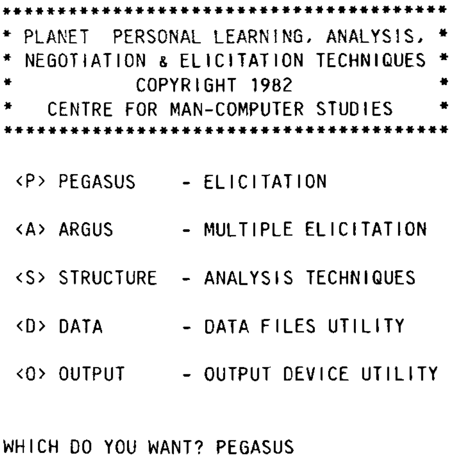
SOCIOGRIDS analyses a set of repertory grids elicited from a group who share the same elements. It compares every pair of grids using the FOCUS algorithm and produces a set of *socionets* which indicates the links of similar construing within the group. A *mode grid* is also produced showing the constructs 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 which has been used is listed in order of how highly matched it is across all grids, and indicating 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.

Some very practical details

The initial experience of sending out the Apple II version of PLANET to a number of unsupervised users led to the addition of a range of features which were inessential to the actual repertory grid application yet vital to its practical use. For example some of the common operational errors were for the data disk to be absent, or the wrong one used, or for the printer to be left off-line so that the system hung up waiting for it to be available. Hence, explicit checks were built in before any programs using either the data disk or printer were run that both were correctly available. Such features may seem trivial, yet they are not ones that are needed for experienced users and it is not at all obvious in advance that they are necessary. It also requires some attention to human factors to ensure that such checks occur at places that are natural to the user and where the reason for them, and the remedy to any problem, are both obvious.

Over a period of time all the issued programs were integrated through a branching sequence of standardized *menus* that offered similar facilities in a uniform fashion. Figure 4 shows the main PLANET menu as it appears on the screen.



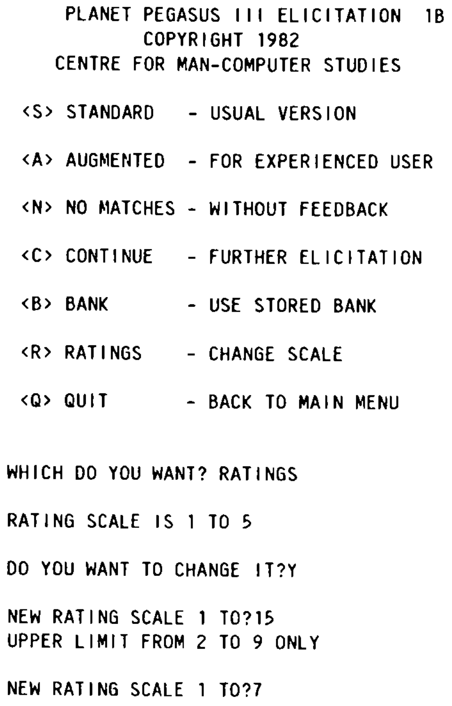
```
*************************************
* PLANET  PERSONAL LEARNING, ANALYSIS, *
* NEGOTIATION & ELICITATION TECHNIQUES *
*             COPYRIGHT 1982           *
*   CENTRE FOR MAN-COMPUTER STUDIES    *
*************************************

 <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? PEGASUS
```

FIG. 4. PLANET Initial Menu. The user has typed "P" to select the PEGASUS program. The system fills in the remainder and then transfers control to the PEGASUS menu.

Key words were chosen for action through the menus such that the user only had to type the initial letter of a request and the system could fill in the rest.

Operational error problems occurred at a variety of levels and due to a variety of factors. Checks were built in at nodes where the user could enter data or parameters. For example, Fig. 5 shows an error feedback sequence when the user is changing the rating scale range to be used by PEGASUS.

Also, Applesoft BASIC does not distinguish between creating a new file and opening an existing one. Hence it is important to check on the one hand whether a data file



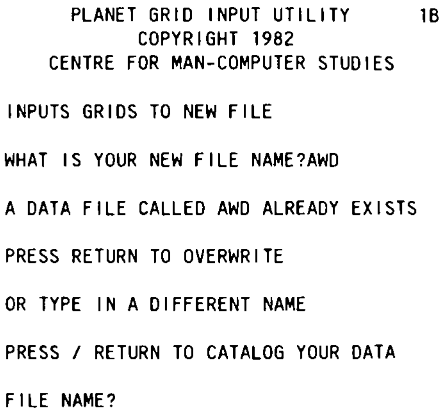
```
          PLANET PEGASUS III ELICITATION  1B
                   COPYRIGHT 1982
          CENTRE FOR MAN-COMPUTER STUDIES

      <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? RATINGS

    RATING SCALE IS 1 TO 5

    DO YOU WANT TO CHANGE IT?Y

    NEW RATING SCALE 1 TO?15
    UPPER LIMIT FROM 2 TO 9 ONLY

    NEW RATING SCALE 1 TO?7
```

FIG. 5. PLANET PEGASUS III Menu. The user has typed "R" to select a change in the rating scale range. He attempts to change it outside the available range and the system gives a helpful response. He changes it to 1–7 and the system transfers control back to the PEGASUS III menu.

to be processed exists before attempting to use it—the user may have typed the wrong name and need to be informed of this; on the other hand if the user is creating a new file and an old one of the same name exists then again this action needs to be queried and the opportunity to catalogue files on the system should be given. Figure 6 shows such a dialogue sequence at the beginning of the grid INPUT utility.

Such measures can be taken when errors are expected to occur. However, not all possible operational errors can be foreseen and to catch unexpected problems the DOS 3.3 operating system was modified so that any form of error led to a system re-boot, as did pressing the Apple II RESET key. Thus, the user could never be left in an ill-defined state with no access to system utilities. So much error-trapping has now been built in to all the programs that such re-booting has become difficult to activate and PLANET usually responds with a diagnostic message, but the re-boot is the ultimate salvation for totally unexpected situations. In retrospect such features are clearly necessary, and it is possible to give general prescriptions for programming good interactive dialogue (Gaines, 1981; Gaines & Shaw, 1982), but anyone who attempts to issue software developed in a research environment to a wide range of remote users should not underestimate the difficulties of making the system user-supportive.

Another significant problem area has been that of the wide variety of printers and printer protocols on the Apple II. Whereas the machine configuration, operating

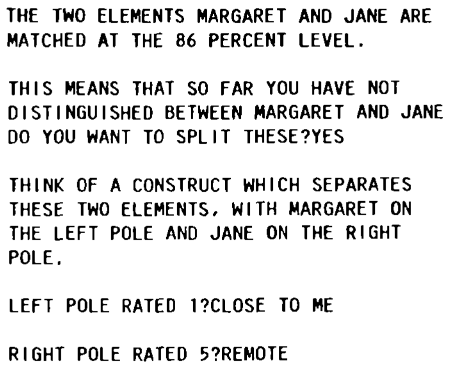


```
                PLANET GRID INPUT UTILITY       1B
                      COPYRIGHT 1982
              CENTRE FOR MAN-COMPUTER STUDIES

         INPUTS GRIDS TO NEW FILE

         WHAT IS YOUR NEW FILE NAME?AWD

         A DATA FILE CALLED AWD ALREADY EXISTS

         PRESS RETURN TO OVERWRITE

         OR TYPE IN A DIFFERENT NAME

         PRESS / RETURN TO CATALOG YOUR DATA

         FILE NAME?
```

FIG. 6. PLANET INPUT Utility Initial Dialogue. The user gives AWD as the name of a new data file of grids to be created. The system notes that a file with this name already exists and gives the user the additional options of changing the name or cataloging the data files rather than just overwriting the existing file.

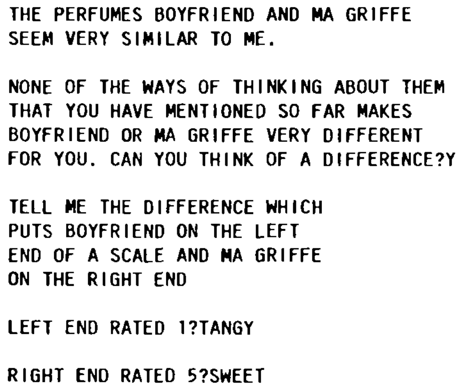
system and language is more highly standardized than that of most microcomputers, the printers used vary considerably. Since one of the key features of the analysis programs in PLANET is the meaningful layouts of the clustered data, exact printer control is very important. To solve this problem a general interpreter was written which handles all printer setting-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 through the OUTPUT utility in PLANET, 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 also allows messages to be put out on the screen as to which printer is being used and when. This facility can also be used by those having an 80-column card in a different peripheral slot.

Having achieved a high degree of operator independence and printer independence it was not surprising to be asked for some degree of language independence in the dialogue. English has long dominated the programming language vocabulary but as personal computers come into use with a wide range of users of differing nationalities it is necessary to allow for the dialogue to be re-written as required. This is particularly significant for the interactive grid elicitation program, PEGASUS, since this elicits purpose, constructs and elements from users, and these can be in any language. However, the PEGASUS feedback about relationships between the elicited material needs to be in the same language also, as does its general instructions to the user. PEGASUS is a complex program with many different paths and options at differing stages of elicitation and re-writing it for each different language was unrealistic. Hence, all the dialogue was stripped out of the program and kept in a separate file read up at run time. This feature has also been used in some specialist applications to tailor the very general PEGASUS dialogue more closely to the problem. For example, Fig. 7 shows part of the dialogue following an element match in the general case, and Fig. 8 shows it in a specific market research application.



```
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. 7. Element match feedback dialogue from standard PLANET dialogue file.



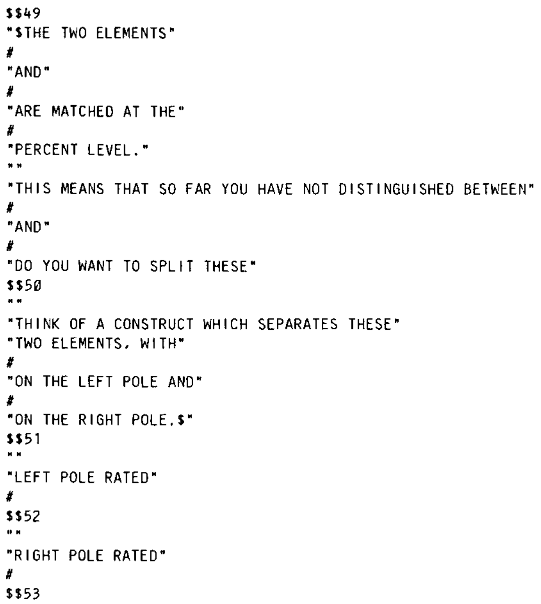
```
THE PERFUMES BOYFRIEND AND MA GRIFFE
SEEM VERY SIMILAR TO ME.

NONE OF THE WAYS OF THINKING ABOUT THEM
THAT YOU HAVE MENTIONED SO FAR MAKES
BOYFRIEND OR MA GRIFFE VERY DIFFERENT
FOR YOU. CAN YOU THINK OF A DIFFERENCE?Y

TELL ME THE DIFFERENCE WHICH
PUTS BOYFRIEND ON THE LEFT
END OF A SCALE AND MA GRIFFE
ON THE RIGHT END

LEFT END RATED 1?TANGY

RIGHT END RATED 5?SWEET
```

FIG. 8. Element match feedback dialogue from specialist PLANET used in a market research study.

An interpreter was again written so that the dialogue file could contain instructions for screen control, and allow parameters to be inserted within the dialogue from the program, such as the construct names. The interpreter also acts as a word-processor formatting the text for the screen width chosen so that the dialogue itself can be written in a fairly free form. Figure 9 shows part of the dialogue file which generated the text of Fig. 7, and Fig. 10 shows the equivalent part generating the dialogue of Fig. 8.

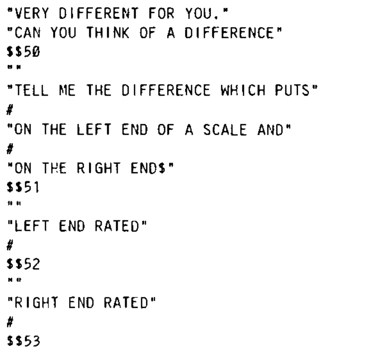
Conclusions

The low-cost microcomputer has made feasible approaches to automated psychological testing which would have been totally unrealistic a decade ago. Not only is it possible to provide automatic administration of the tests themselves, but it is also possible to integrate them into a system that provides a wide range of supporting facilities to the user. In this paper I have described how a set of repertory grid elicitation and analysis programs originally developed on a large mainframe and widely used on a variety of systems has been transported and greatly extended on a microcomputer.



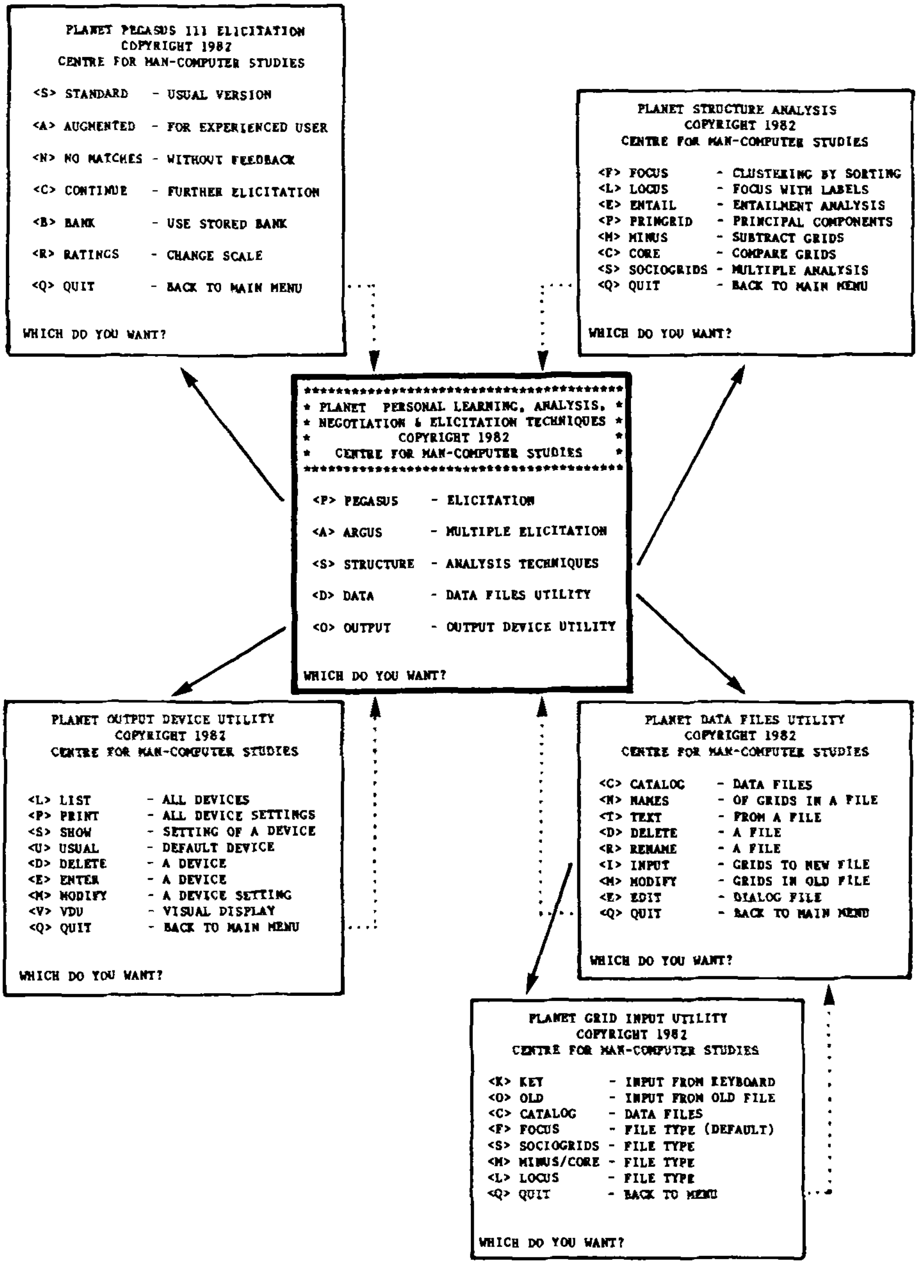
```
$$49
"$THE TWO ELEMENTS"
#
"AND"
#
"ARE MATCHED AT THE"
#
"PERCENT LEVEL."
" "
"THIS MEANS THAT SO FAR YOU HAVE NOT DISTINGUISHED BETWEEN"
#
"AND"
#
"DO YOU WANT TO SPLIT THESE"
$$5Ø
" "
"THINK OF A CONSTRUCT WHICH SEPARATES THESE"
"TWO ELEMENTS, WITH"
#
"ON THE LEFT POLE AND"
#
"ON THE RIGHT POLE.$"
$$51
" "
"LEFT POLE RATED"
#
$$52
" "
"RIGHT POLE RATED"
#
$$53
```

FIG. 9. The section of the PLANET dialogue file generating the text of Fig. 7. The $$49, $$50, and so on, are delimiters separating sections of dialogue. The # indicates that a parameter to be substituted will be passed to the dialogue interpreter, such as the element names and the match level. A $ at the beginning of line causes the screen to be cleared. A $ at the end of a line causes a shift to the beginning of the next line on the screen.



```
$$49
"$THE PERFUMES"
#
"AND"
#
"SEEM VERY SIMILAR TO ME.
" "
"NONE OF THE WAYS OF THINKING ABOUT THEM"
"THAT YOU HAVE MENTIONED SO FAR MAKES"
#
"OR"
#
"VERY DIFFERENT FOR YOU."
"CAN YOU THINK OF A DIFFERENCE"
$$5Ø
" "
"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. 10. The section of the PLANET dialogue file generating the text of Fig. 8. This dialogue is used by the interpreter in the same way as that of Fig. 9. 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.



```
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?
```

```
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?
```

```
*****************************************
* 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?
```

```
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?
```

```
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?
```

```
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. 11. PLANET System. The overall flow of control in PLANET may be seen through the linkages between the six main menus.

Figure 11 shows the overall PLANET system through printouts of the menus as they appear on the screen together with the flow of control between them. The system provides: multiple facilities for eliciting grids through conversational interaction (PEGASUS and ARGUS); multiple facilities for analysing individual grids (FOCUS, LOCUS, ENTAIL and PRINGRID); multiple facilities for analysing pairs and groups of grids (MINUS, CORE and SOCIOGRIDS); a utility for manipulating the database of grid files on the user's disk (DATA); a utility for creating new grid files from the keyboard or from those already in the database (INPUT); and a utility for operating a variety of printers in different modes (OUTPUT). It also allows the dialogue structure for conversational grid elicitation in PEGASUS to be modified for different purposes and put into different languages.

I have concentrated in this paper not on the main repertory grid programs which have been described in detail elsewhere (Shaw, 1980, 1981), but rather on the different approach that had to be taken to them and their support when they were re-written as an integrated system to be issued widely to remote users with no direct contact with the originator. The experience of doing this illuminates many of those points made by others considering features of microcomputer systems relevant to automated testing (Beaumont, 1982) and I hope the problems discovered and techniques adopted to overcome them will be useful to others also attempting to exploit the very great potential that low-cost microcomputers have for automated psychological testing.

I am very grateful to my colleague and husband, Brian Gaines, for his help in implementing the PLANET system on the Apple II computer. I am also grateful to a number of other colleagues and users who have patiently reported problems and helped to check thoroughly the operation of PLANET.

References

ACKER, W. (1982). A computerized approach to psychological screening—the Bexley-Maudsley Automated Psychological Screening and the Bexley-Maudsley Category Sorting Text. *International Journal of Man-Machine Studies*, **17**, 361–369.

BEAUMONT, J. G. (1982). System requirements for interactive testing. *International Journal of Man-Machine Studies*, **17**, 311–320.

CALVERT, E. J. & WATERFALL, R. C. (1982). A comparison of conventional and automated administration of Raven's Standard Progressive Matrices. *International Journal of Man-Machine Studies*, **17**, 305–310.

CARD, W. I., NICHOLSON, M., CREAN, G. P., WATKINSON, G., EVANS, C. R., WILSON, J. & RUSSELL, D. (1974). A comparison of doctor and computer interrogation of patients. *International Journal of Biomedical Computing*, **5**, 175–187.

ELITHORN, A. & TELFORD, A. (1969). Computer analysis of intellectual skills. *International Journal of Man-Machine Studies*, **1**, 189–209.

ELWOOD, D. L. (1972*a*). Test retest reliability and cost analyses of automated and face to face intelligence testing. *International Journal of Man-Machine Studies*, **4**, 1–22.

ELWOOD, D. L. (1972*b*). Automated WAIS testing correlated with face-to-face testing: a validity study. *International Journal of Man-Machine Studies*, **4**, 129–137.

ELWOOD, D. L. (1972*c*). Automated versus face-to-face intelligence testing: comparison of test-retest reliabilities. *International Journal of Man-Machine Studies*, **4**, 363–369.

FRANSELLA, F. & BANNISTER, D. (1977). *A Manual for Repertory Grid Technique*. London: Academic Press.

GAINES, B. R. (1981). The technology of interaction—dialogue programming rules. *International Journal of Man-Machine Studies*, **14**, 133–150.

GAINES, B. R. & SHAW, M. L. G. (1982). The human protocol. *Proceedings of Compcon 82*. Washington, D.C.: IEEE Computer Society.

GEDYE, J. L. & MILLER, E. (1969). The automation of psychological assessment. *International Journal of Man-Machine Studies*, **1**, 237–262.

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

KLINGE, V. & RODZIEWICZ, T. (1976). Automated and manual intelligence testing of the Peabody Picture Vocabulary Test on a psychiatric adolescent population. *International Journal of Man–Machine Studies*, **8**, 243–246.

LUCAS, R. W. (1977). A study of patients' attitudes to computer interrogation. *International Journal of Man–Machine Studies*, **9**, 69–86.

RIDGWAY, J., MACCULLOCH, M. J. & MILLS, H. E. (1982). Some experiences in administering a psychometric test with a light pen and microcomputer. *International Journal of Man–Machine Studies*, **17**, 265–278.

SHAW, M. L. G. (1980). *On Becoming a Personal Scientist*. London: Academic Press.

SHAW, M. L. G., Ed. (1981). *Recent Advances in Personal Construct Technology*. London: Academic Press.

SLATER, P., Ed. (1976). *Dimensions of Intrapersonal Space*, vol. 1. London: John Wiley.

SLATER, P., Ed. (1977). *Dimensions of Intrapersonal Space*, vol. 2. London: John Wiley.

STODOLSKY, D. (1970). The computer as a psychotherapist. *International Journal of Man–Machine Studies*, **2**, 327–350.

THOMPSON, J. A. & WILSON, S. L. (1982). Automated Psychological Testing. *International Journal of Man–Machine Studies*, **17**, 279–289.

VOLANS, P. J. & LEVY, R. (1982). A re-evaluation of an automated tailored test of concept learning with elderly psychiatric patients. *British Journal of Clinical Psychology*, **21**, 39–47.

WATTS, K., BADDELEY, A. & WILLIAMS, M. (1982). Automated tailored testing using Raven's Matrices and the Mill Hill Vocabulary tests: a comparison with manual administration. *International Journal of Man–Machine Studies*, **17**, 331–344.