

# A Computer Aid to Knowledge Engineering

Mildred L. G. Shaw & Brian R. Gaines

Department of Computer Science, York University,  
4700 Keele Street, Downsview, Ontario, Canada M3J 1P3

&

Department of Industrial Engineering  
University of Toronto, Ontario, Canada M5S 1A4

Advances in the technology of expert systems have made it possible to implement and operate cost-effective systems in complex domains of great practical importance. However, the knowledge engineering necessary to set up an expert system in a new domain is currently labour-intensive and requires skilled staff to elicit and codify the knowledge of an expert. This paper outlines a methodology for knowledge elicitation that has been implemented in a computer program that interacts with an expert to enable him to express the constructs underlying his knowledge. Some preliminary experiments on validating the system by using it to re-construct the *Business Information Analysis and Integration Technique* are outlined, and directions for further empirical studies of the elicitation of accountancy expertise are indicated.

## Introduction

The initial success of expert system developments (Michie, 1979) 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 usage of expert systems. However, what Feigenbaum (1980) termed *knowledge engineering*, the reduction of a large body of knowledge to a precise set of facts and rules, has already become a major bottleneck impeding the application of expert systems in new domains. We need to understand more about the nature of expertise in itself (Hawkins, 1983) and to be able to apply this knowledge to the elicitation of expertise in specific domains.

The problem has been stated very 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 et al., 1983, p.129)*

In this paper we describe a technique to overcome these communication problems and aid the knowledge engineer in developing the expert’s vocabulary. The expert interacts with a computer program, PEGASUS, that encourages him to make clear the distinctions he uses in applying his expertise. This helps him to structure his knowledge and identify and formalize his concepts. PEGASUS may also be used in a teaching mode in order to enable others to come to use the expert’s vocabulary in the same way as he does. We describe some preliminary experiments to

validate the use of PEGASUS in this way by re-constructing the basic distinctions used in the Business Information Analysis and Integration Technique (BIAIT) used to determine the accounting needs of a company (Carlson, 1979; Sowa, 1984, pp.301-302).

## **Construct Elicitation**

PEGASUS (Shaw, 1980b) is based on a theory of human cognition developed by Kelly (1955) termed *personal construct theory* and applied initially in clinical psychology and later in education and management. Kelly developed a systemic theory of cognition based on the single primitive of a *construct*, or dichotomous distinction. He proposes that all of human activity can be seen as a process of anticipating the future by construing the replication of events. Hence his psychological model of man is strongly epistemological and concerned with the way in which man models his experience and uses this model to anticipate the future. The anticipation may be passive, as in prediction, or active, as in action.

Kelly's explication of man's modelling process is extraordinarily simple. Personal constructs are dichotomous, or bipolar, distinctions. Constructs themselves are construed and hence form a hierarchy or *construct system*. At the upper levels of the hierarchy there tend to be evaluative constructs that form the basis of appraisal of events, manifesting themselves in emotions and the goals for action. At the lower levels there tend to be ways of classifying experience. Kelly emphasizes the highly *dynamic* and *personal* nature of the construct system. It is subject to change according to feedback from failure to anticipate events. It is individualistic in the constructs used, in the vocabulary used to name the construct poles, in the relations between constructs in the hierarchy, and in those constructs most likely to change when this is necessary.

Knowledge engineering for expert systems may be seen as a process of making overt the construct system of the expert. The problems noted in the previous section arise because of the essentially individualistic nature of construct systems. A person may be able to use his construction system effectively whilst having no basis for communicating it to others. Two people may use exactly the same construct yet refer to its poles by different names. Two people may use the same names for the poles of their constructs and yet use them in different ways. Two people may use similar constructs at the lower level of the hierarchy and yet have them organized in different systems such that their reactions to the same event are quite different. Two people may have similar constructs at nearly all levels of the hierarchy and yet construe a novel event completely differently.

In a number of past papers we have given an exposition of construct theory (Shaw, 1980b; Shaw and Gaines, 1981a), developed its systemic foundations (Gaines and Shaw, 1981), given it a fuzzy semantics (Shaw and Gaines, 1982; Shaw and Gaines, 1984), shown how communal knowledge arises (Shaw and Gaines, 1981c), described algorithms for the analysis of construct systems (Shaw, 1980b; Shaw and Gaines, 1981b), discussed their implementation in the PLANET suite of computer programs (Shaw, 1982) and described applications of the techniques and programs in psychology (Shaw, 1981), decision making (Shaw and Gaines, 1981b; Shaw and Gaines, 1984) and systems analysis (Shaw and Gaines, 1983). In the following section we take an example from the expert systems literature and show the application of personal construct psychology to knowledge engineering.

## Eliciting the Constructs Underlying Knowledge

One of the problems of evaluating any methodology for knowledge engineering is to have some clear-cut test cases where the conceptual framework of experts has been both elicited and validated. Sowa (1984) describes one such case where the record keeping needs of a business 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 our personal construct elicitation methodology operationalized through the PEGASUS program. The key question is:

*Q1. Can PEGASUS elicit the seven BIAIT constructs?*

This is not as straightforward as it appears since to elicit constructs we have to give *elements*, that is examples to which they can be applied. A standard technique for construct elicitation is to give a triad of elements and ask the question, “In what way are two of these alike and different from the third?” PEGASUS uses this as one method of eliciting an expert’s constructs. The program may start with a set of pre-defined elements or allow the user to define all the elements. It seems likely that the initial choice of elements may effect the constructs elicited and hence the question needs splitting:

*Q1.1. Can PEGASUS elicit the seven BIAIT constructs starting with a defined set of elements?*

*Q1.2. Can PEGASUS elicit the seven BIAIT constructs starting with no elements defined?*

A second dimension of variation in the problem is whether we elicit the constructs from an expert such as an accountant, or from non-experts. It is an interesting question always as to whether the constructs in use by experts are in the public domain and it is their application that embodies expertise, or whether the constructs themselves are specialized.

*Q1.1.1. Can PEGASUS elicit the seven BIAIT constructs from an expert starting with a defined set of elements?*

*Q1.1.2. Can PEGASUS elicit the seven BIAIT constructs from a non-expert starting with a defined set of elements?*

*Q1.2.1 Can PEGASUS elicit the seven BIAIT constructs from an expert starting with no elements defined?*

*Q1.2.2 Can PEGASUS elicit the seven BIAIT constructs from a non-expert starting with no elements defined?*

These questions seem ordered in decreasing likelihood of success in that if the answer to Q1.1.1 is “no” then so it will be for the others. We have started a program of research to answer these questions empirically through a series of controlled experiments with accountants, businessmen and the general public as subjects. The next section reports an initial informal run to show PEGASUS in action.

## **Eliciting the BIAIT Constructs**

PEGASUS commences its interaction by asking for the purpose of the elicitation and then for a set of elements relevant to this purpose. This run is concerned with Q1.1.1 and all these were supplied to the businessman using PEGASUS (input shown in **bold**):

Type in your purpose for doing this grid? **to analyse customer-supplier relationships from an accounting point of view**

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.

Element 1? **customer of Safeways**

Element 2? **customer of Hertz**

Element 3? **customer of Consumer Gas**

Element 4? **customer of Custom Drapes**

Element 5? **customer of real-estate firm**

Element 6? **customer of dentist**

One technique used by PEGASUS to elicit constructs is that of triadic comparison:

Triad for elicitation of construct 1

1 customer of Safeways

2 customer of Hertz

3 customer of Consumer Gas

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? **2**

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 customer of Safeways and customer of Consumer Gas on the left pole from customer of Hertz 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 --? **buy**

right pole rated 5 --? **rent**

Now assume that customer of Safeways and customer of Consumer Gas are assigned the value 1 and customer of Hertz 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

1 customer of Safeways 1  
3 customer of Consumer Gas 1  
2 customer of Hertz 5  
4 customer of Custom Drapes ? 1  
5 customer of real-estate firm ? 1  
6 customer of dentist ? 1

Another technique is that of matching the ratings given to elements and asking for a construct that differentiates between two highly matched elements:

The two elements 3 customer of Consumer Gas and 6 customer of dentist are matched at the 87 percent level. This means that so far you have not distinguished between customer of Consumer Gas and customer of dentist.

Do you want to split these? **Y**

Think of a construct which separates these two elements, with customer of Consumer Gas on the left pole and customer of dentist on the right pole

left pole rated 1 --? **paid by customer**  
right pole rated 5 --? **paid by another**

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

3 customer of Consumer Gas 1  
6 customer of dentist 5  
1 customer of Safeways ? 1  
2 customer of Hertz ? 2  
4 customer of Custom Drapes ? 1  
5 customer of real-estate firm ? 1

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:

The two constructs you called

2 one-off sales -- long-term service  
6 goods -- services

are matched at the 71 percent level

This means that most of the time you are saying one-off sales you are also saying goods and most of the time you are saying long-term service you are also saying services. Think of another element which is either one-off sales and services or goods and long-term service. 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? **customer of shoe-shine boy**

**Type in the ratings for this element on each construct.**

Left pole rated 1, right pole rated 5.

buy--rent? 1  
one-off sales--long-term service? 1  
price fixed--price negotiable? 2

standard--special? 1  
 paid by customer--paid by another? 1  
 goods--services? 5

Through a combination of triadic comparison, element matching, and construct matching, a PEGASUS interaction draws out from the user his conceptual framework relating to the purpose of the elicitation. The result is what Kelly calls a *repertory grid* of elements rated on constructs. PLANET contains a variety of techniques for analysing such grids, including entailment analysis and hierarchical clustering. However, for the purposes of this paper it is sufficient to examine the resulting grid sorted by similar elements and constructs:

|                      |       | * 2   | 8     | 1     | 7     | 4     | 9     | 5     | 3     | 6     |                              |
|----------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------------------------------|
|                      | ***** |       |       |       |       |       |       |       |       |       |                              |
| consumable           | 9     | * 1   | 1     | 1     | 5     | 5     | 5     | 5     | 1     | 4     | durable                      |
| <b>price fixed</b>   | 3     | * 1   | 2     | 1     | 1     | 1     | 5     | 5     | 1     | 1     | <b>price negotiable</b>      |
| one payment          | 7     | * 1   | 1     | 1     | 3     | 1     | 1     | 5     | 1     | 1     | stage payments               |
| <b>standard</b>      | 4     | * 1   | 1     | 1     | 1     | 5     | 1     | 1     | 1     | 2     | <b>special</b>               |
| <b>buy</b>           | 1     | * 5   | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     | <b>rent</b>                  |
| paid by customer     | 5     | * 2   | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 5     | paid by another              |
| <b>one-off sales</b> | 2     | * 2   | 1     | 1     | 1     | 1     | 1     | 2     | 4     | 5     | <b>long-term service</b>     |
| goods                | 6     | * 4   | 5     | 1     | 1     | 1     | 1     | 2     | 2     | 5     | services                     |
| <b>delivered</b>     | 11    | * 4   | 5     | 5     | 1     | 4     | 3     | 3     | 1     | 5     | <b>collected</b>             |
| <b>billed</b>        | 12    | * 3   | 5     | 5     | 4     | 3     | 4     | 1     | 1     | 3     | <b>pay cash</b>              |
| cash                 | 8     | * 5   | 1     | 3     | 4     | 4     | 3     | 1     | 1     | 1     | credit card                  |
| necessity            | 10    | * 4   | 5     | 1     | 1     | 4     | 5     | 1     | 1     | 1     | luxury                       |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * |                              |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of dentist          |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of Consumer Gas     |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of real-estate firm |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of antique store    |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of Custom Drapes    |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of furniture store  |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of Safeways         |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of shoe-shine boy   |
|                      |       | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | * * * | customer of Hertz            |

The BIAIT constructs that have been elicited are shown in **bold**. There is one missing, number 6, “does the supplier keep track of the product after it is sold?” There are no elements corresponding where the supplier has a continuing product responsibility in this grid, probably because the examples have been drawn from consumer rather than professional markets. This shows the importance of having a sufficient range of elements to elicit the full dimensions of the conceptual structure.

Six non-BIAIT constructs have also been elicited and, if they are regarded as giving irrelevant distinctions, it is important to consider how they may be removed. The grid is used for discussion purposes and the next question to be asked is how an order being at one pole or the other of a construct affects the record keeping required. This removes constructs 9, 6 and 10 which relate to the nature of the business, not its record keeping. Construct 7 turns out to be a sub-case of the missing BIAIT construct and may lead to it under discussion. Constructs 5 and 8 are sub-cases of construct 12, where the billing is to a third party.

The program ENTAIL (Gaines and Shaw, 1980) gives a dependency analysis of the grid by listing logical entailments not falsified by the data and it was expected that this would aid

removal of unnecessary constructs. However, this is only partially successful because the grid as elicited contains mixed cases which show up as non-extreme ratings (2,3,or 4). For example, some customers of Hertz pay cash and others by credit card. When these customers are rated on other constructs this information is lost. We are extending the interaction to query the rating of elements that are not at extremes. The user will be given the opportunity to describe the elements in more detail as short case histories (i.e. instantiated case frames) and to add qualifications to these as he construes them in relation to new constructs.

We are also extending the dependency analysis to be interactive so that it may lead to the elicitation of further elements if the system suggests dependencies with which the expert disagrees, or to the modification of ratings if the expert suggests dependencies which are not possible according to the data he has entered. The overall objective is to allow the expert to explore the dimensions of his conceptual structure, testing it against specific cases, and discussing its implications, until he is comfortable that the representation elicited reasonably represents his knowledge.

The system we have described is concerned primarily with the expert's conceptual structure for describing problems. It can also be applied to his conceptual structure for describing solutions or actions, in this case the different forms of record keeping appropriate to different types of business. Rule elicitation then consists of determining the relationship between these two conceptual spaces. PEGASUS already has provision for enabling one user (the learner) to obtain feedback on the relationship of his constructs to those of another (the expert), so that it is possible to use the same program and techniques to train users of an expert system in the proper use of the expert's terminology (Shaw, 1980a).

## **Summary and Conclusions**

An interactive computer aid to expedite the initial stages of knowledge engineering has been described and illustrated through an example. The PEGASUS program elicits from an expert the fundamental distinctions, or constructs, he is using in applying his expertise. The ENTAIL program analyses the dependencies between these constructs from the data he supplies. An initial informal trial of the techniques in a test case based on the BIAIT analysis of the record-keeping requirements of business enterprises shows some promise.

We are extending the programs currently to allow further discussion with experts on their conceptual structures and commencing a series of formal experiments to evaluate the technique. We are also examining our other areas of study, expert systems in agriculture and the oil industry in Canada, for similarly well-defined constructs to those of BIAIT. It seems particularly important at this stage in the development of expert systems and knowledge engineering to establish some standards by which various heuristic techniques may be judged. The methodology described in this paper suggests one form of normative testing environment.

## **Acknowledgements**

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

## References

- Carlson, W.M. (1979). The new horizon in business information analysis. **Data Base** 10(4) 3-9.
- Feigenbaum, E. (1980). Knowledge Engineering: the Applied Side of Artificial Intelligence. STAN-CS-80-812 Department of Computer Science, Stanford University.
- Gaines, B.R. and Shaw, M.L.G. (1980). New directions in the analysis and interactive elicitation of personal construct systems. **International Journal Man-Machine Studies** 13 81-116.
- Gaines, B.R. and Shaw, M.L.G. (1981). A programme for the development of a systems methodology of knowledge and action. Reckmeyer, W.J., Ed. **General Systems Research and Design: Precursors and Futures**. pp.255-264. Louisville, Kentucky, Society for General Systems Research.
- Hawkins, D. (1983). An analysis of expert thinking. **International Journal of Man-Machine Studies** 18(1) 1-47.
- Hayes-Roth, F., Waterman, D.A. and Lenat, D.B., Ed. (1983). **Building Expert Systems**. Reading, Massachusetts, Addison-Wesley.
- Kelly, G.A. (1955). **The Psychology of Personal Constructs**. New York, Norton.
- Michie, D., Ed. (1979). **Expert Systems in the Micro Electronic Age**. Edinburgh, Edinburgh University Press.
- Shaw, M.L.G. (1980a). Acquisition of personal knowledge. **International Journal of Policy Analysis and Information Systems** 4(4) 401-429.
- Shaw, M.L.G. (1980b). **On Becoming A Personal Scientist: Interactive Computer Elicitation of Personal Models Of The World**. London, Academic Press.
- Shaw, M.L.G., Ed. (1981). **Recent Advances in Personal Construct Technology**. London, Academic Press.
- Shaw, M.L.G. (1982). PLANET: some experience in creating an integrated system for repertory grid applications on a microcomputer. **International Journal of Man-Machine Studies** 17(3) 345-360.
- Shaw, M.L.G. and Gaines, B.R. (1981a). Exploring personal semantic space. Rieger, B., Ed. **Empirical Semantics II**. pp.712-791. Bochum, West Germany, Studienverlag Brockmeyer.
- Shaw, M.L.G. and Gaines, B.R. (1981b). The personal computer and the personal scientist. **BCS'81: Information Technology for the Eighties**. pp.235-252. London, Heyden Press for British Computer Society.
- Shaw, M.L.G. and Gaines, B.R. (1981c). The personal scientist in the community of science. Reckmeyer, W.J., Ed. **General Systems Research and Design: Precursors and Futures**. pp.59-68. Kentucky, Society for General Systems Research.
- Shaw, M.L.G. and Gaines, B.R. (1982). Eliciting entailment. Trappl, R., Ricciardi, L. and Pask, G., Ed. **Progress in Cybernetics and Systems Research. Vol. IX**. pp.425-435. Washington, Hemisphere.
- Shaw, M.L.G. and Gaines, B.R. (1983). Eliciting the real problem. Wedde, H., Ed. **Adequate Modeling of Systems**. pp.100-111. Berlin, Springer.
- Shaw, M.L.G. and Gaines, B.R. (1984). Deriving the constructs underlying decision. Zimmermann, H.J., Zadeh, L.A. and Gaines, B.R., Ed. **Fuzzy Sets and Decision Analysis. TIMS Series Studies in the Management Sciences, 20**. pp.335-355. Amsterdam, North-Holland.
- Sowa, J.F. (1984). **Conceptual Structures: Information Processing in Mind and Machine**. Reading, Massachusetts, Addison-Wesley.