Coping with a Poorly Understood Domains: The Example of Internet Trust

Andrew Basden *The Centre for Virtual Environments, University of Salford.*

John B. Evans, David W. Chadwick, Andrew Young *Information Technology Institute, University of Salford.*

ABSTRACT

The notion of trust, as required for secure operations over the Internet, is important for ascertaining the source of received messages. How can we measure the degree of trust in authenticating the source? Knowledge in the domain is not established, so knowledge engineering becomes knowledge generation rather than mere acquisition. Special techniques are required, and special features of KBS software become more important than in conventional domains. This paper generalizes from experience with Internet trust to discuss some techniques and software features that are important for poorly understood domains.

Keywords: Internet trust, knowledge-based tools, knowledge elicitation, knowledge-poor domains, knowledge refinement, knowledge generation, Istar.

1. INTRODUCTION

1.1 Knowledge Generation

The traditional view of knowledge acquisition is linear (Fig. 1a), in which a knowledge engineer extracts pieces of knowledge from a source and then represents them by symbols in the knowledge representation language. But such an approach assumes the pieces of knowledge already exist in more-or-less finished form and merely need uncovering. When this view is applied to tacit knowledge (1) its explication is seen in terms of merely clearing away the layers that hide the knowledge until it is brought to light.

But in many domains this view is inappropriate. Knowledge is not just extracted but also generated, created, formed by the process of knowledge acquisition and knowledge representation. The process gains a circular element, Fig. 1b, in which the act of representing the knowledge stimulates the thinking of the knowledge engineer to think of new pieces and to refashion existing pieces.

|-|IG "r:kgref/ES98 Fig 1" -w5.33 -h4.15 -ra

|-|CE Fig. 1. Linear and Circular Knowledge Acquistion

This process has been discussed in some depth (2), showing it to be different to the conventional process in many ways; there the two approaches are called 'assembly' and 'creative design'. These differences, which are summarised in Table 1, are fundamental, not just small variations in technique, and thus require a rethinking of knowledge acquisition techniques.

They also mean that the software features found in KBS software are seen in a different light, with different criteria becoming important. Traditionally, such things as expressive power have been assumed to be the most important features; in domains where knowledge is poorly understood and must be generated, expressive power takes second place to whether the use of the software is 'proximal' or 'distal' (3) and how easy it is for the knowledge engineer to change knowledge rather than merely add to it. Features that aid interpretation of ill-defined concepts, both by knowledge engineer and by the end user, assume an importance far greater than in many conventional knowledge domains.

In this paper we discuss the attempt to construct a knowledge based system in such a poorly understood domain, that of Internet trust. After a brief description of the project, we discuss four different sources of the need for knowledge generation, then describe our experiences during the project. The discussion then generalises these, and yields two lists of tentative recommendations for knowledge acquisition methodology and for software features needed for exercises of this kind.

|-|CE TABLE 1. Comparison of knowledge engineering paradigms

Paradigm	Assembly	Creative design
KBS purpose Advice-giving Knowledge Refinement Training		
Knowledge source External The process itself		
KR Tool users Single Group		
KR Tool purpose Representation only Communication Clarification Representation		
Nature of kg. eng. Monotonic increment Non-monotonic reinterpretation Planned or situated actions Continuous process Evolutionary Revolutionary		
Prime symbol level task Add pieces of knowledge Change knowledge		
Style of user action Discrete user tasks Continuous activity Predefined goals Often no task goals Means to end Means and ends not separated 'How' unimportant 'How' important Actions result in Some actions without internal changes internal result		

Tool-user Relationship Distal Proximal

Quality criteria Learnability Proximality Standardization Lack of interruption

Style of user interface Object Oriented Holistic

1.2 The Intelligent Computation of Trust (ICT) Project

In order to perform important (e.g. commercial) transactions over the Internet it is essential that the parties are sure of each other's true identity. Clearly before anyone would be prepared to invest time, effort and possibly resources into a transaction it would be wise to firstly ensure that the other party is genuine. The first step is to consider the problem of ascertaining the authenticity of a remote party. Commercial reality would demand that the identity should reference an entity that could be sued in court, if necessary.

Not only do we need to know that the document has come from a reliable source (authentication) we also need to know that no-one other than authorised recipients can access the information of the transaction. Encryption is usually used to ensure this confidentiality, involving public keys, but the problem then lies in the secure exchange of the key between the parties. The problem then becomes one of authentication again, i.e. of the public key of an identified entity.

The basic relationship being investigated is a tripartite one where an entity, an individual person or person acting for a commercial company, is assessing the authenticity of (the public keys of) another on the basis of communication over the Internet. The entity turns to a Certification Authority (CA) or Trusted Third Party (TTP) to aid this process. A CA is responsible for authenticating the identity of a subscriber, and for securely binding this identity to the subscriber's public key, through a process known as certification. A relying party can then obtain the certified public key of a subscriber, and providing s/he trusts the binding carried out by the CA, can trust that this really is the public key of the subscriber, and can then with confidence enter into a secure communication with the subscriber.

But it is felt that not all CAs (and there are many of them) are equally good in performing their task. Many CAs have broad responsibilities; besides issuing the keys for encoded communication they will also provide an Internet service as a commercial package. Not only may some be slipshod in their acceptance of subject entities' bona fides, but some may be prone to subversion from either inside the organisation or from hacking operations carried out from the outside. To guard against this a CA publishes a CPS (certification practice statement) which amounts to a statement of the means by which the CA will perform its various duties including the verification of subscribers. Regular compliance audits are made to ensure that the high standards of checking are maintained.

The multifaceted nature of trust leads us to devise a knowledge based system (KBS) to evaluate the claims and attributes of such CAs based on the statements in their CPSs. The main task of such a KBS is to examine (or to guide the user in examining) the CPS published or referenced by the CA to try to estimate the degree of trust a relying party may place upon the recommendations of any particular CA. The approach is to build a model, based on Chokhani and Ford's (4) general framework for CPSs, and then consider to what extent a given CPS measures up to it.

There is, however, very little established high level expertise of what contributes to trust in such a system. It is a highly interpretive notion, highly context dependent and highly volatile in its meaning. Therefore knowledge for such a KB is not ready to hand, and must be generated rather than extracted.

2. TYPES OF KNOWLEDGE GENERATION

When knowledge generation occurs the expert's or user's knowledge is refined and enhanced in such a way that human knowledge increases. Knowledge generation can take place in two ways (5): when using a knowledge based system and when building the knowledge base. Since the ICT KB has not yet been used, this paper discusses the latter. Four types of knowledge generation have been reported.

2.1 Gap-filling

While building an expert system to predict corrosion (6) it was found that during the process of knowledge engineering the expert - a corrosion expert of worldwide renown - would sometimes encounter gaps in his knowledge, which he then filled by performing laboratory experiments. In this way insights gained as a result of the process helped to refine the expert's own knowledge by filling those gaps. Gap filling usually occurs in well established domains like corrosion knowledge; knowledge is probably more refined than generated. An element of this can be found in much knowledge engineering, especially where the domain of knowledge is not well structured.

2.2 Model Contextualisation

Some tasks are not performed routinely, such as crisis management and decision making in emergencies and war time. In such situations, the people who act as knowledge sources might have some general understanding but lack specific experience. There are no experts, and Paul (7) calls such semi-experts 'journs'. He describes the construction of a knowledge base, SARA, by a process of 'knowledge cultivation' which assumes that the existing body of knowledge is incomplete, and proceeds in cycles to build up the knowledge base. Paul calls such domains 'knowledge-poor'.

In these domains the journs often have some model of what should be done, but it lacks detail from the specific context. An important part of the process of knowledge cultivation is therefore the contextualisation of the model by stimulating the journs to consider specific situations (contexts).

2.3 Cognitive Mapping

Strategic group decision making is a third situation in which knowledge generation or refinement is necessary because of knowledge poverty, but the poverty is of a different sort. While many heuristics for decision making are offered by management consultants, each business situation is radically different, as are the participants, so the heuristics often do not apply to any depth and new knowledge has to be developed each time (8).

In these domains a plethora of factors is relevant to the situation, and in each situation a different plethora pertains. So few, if any, real models are available and there is often no previous experience to guide the new situation (or what experience is offered is suspect). A different kind of knowledge base is needed - cognitive maps and influence diagrams (9,10) - which stimulate the participants (often a group) to consider new factors and how they link together in their particular 'plethoric' situation.

2.4 Exploring New Approaches

In the fourth type of knowledge generation, a new knowledge approach is being explored in an established domain. This might occur, for instance, when current practice is being questioned. It is less common than the other types, but was exemplified in the INCA project (2) in which the purpose of the knowledge base was to select clauses for a construction contract. The standard approach to authoring contracts was to make minor amendments to standard forms, but often neither party fully understood the contract and both parties would then seek advantage in an adversarial manner at the end of the construction. The new approach was to author directly from first principles of contract according to what the parties wanted.

When, during knowledge acquisition, the domain experts were asked fundamental questions such as what issues were important in a contract, how to resolve those issues, and how to obtain a balance between the parties to the contract, they could not provide such knowledge, because they seldom considered such questions. What the knowledge engineer was doing was questioning the rationale behind standard procurement methods and standard forms of contract, and little knowledge was available to help him do this.

So the knowledge engineer had to revert to basic principles of the process of procuring a building and of the relationships between actors in the construction process. Once initial principles had been obtained, the knowledge engineer was faced with the tasks of generating knowledge out of those principles, deciding where the links occurred, and generating clausal text for each of the concepts that could be included in contracts. As this progressed, the knowledge itself was frequently refined and modified.

3. THE ICT KNOWLEDGE BASE

3.1 The Domain of Internet Trust

In many ill structured domains, all four of these are present to some extent, but usually one dominates. To build a KB about Internet trust requires some of the fourth type of knowledge generation, but mainly the second type since there is no established body of knowledge and those involved are only 'journs'. Much of what constitutes trust must be worked out from general understanding by considering specific yet hypothetical situations.

3.2 The Istar Software

The Istar software (11) was designed during the INCA project to facilitate the last- mentioned type of knowledge generation: trying a new approach. Its knowledge representation model is reasonably conventional, semantic net with probabilistic inference net, though possessing a rich variety of types of variable. Bayesian variables were the most commonly used in the ICT KB to represent factors that contribute to degree of trust. Each node in the inference net is a variable whose value must be sought from antecedent variables, some of which are questions put to the end user. Variables can either be free variables or attributes of nodes. For an inference session, one or more nodes are designated goals, and a cycle of backward and forward chaining is undertaken until all questions needed to answer the goals have been asked.

Istar's most important characteristic is the highly 'proximal' (3) interface it presents to the knowledge engineer. Donald Norman (12) once remarked,

"The real problem with the interface is that it is an interface. Interfaces get in the way. I don't want to focus my energies on an interface. I want to focus on the job."

The user interface of Istar was designed to allow the knowledge engineer to "get on with the job", so that using the tool could become an integral part of the thinking process. The knowledge engineer draws knowledge on an 'easel' as a box and arrows diagram to express the nodes and arcs of the inference net, rather than entering it either as text or via dialog boxes. Both boxes and arrows are drawn, moved or redirected, with simple press-drag- release mouse movements, without the cognitive load imposed by point-and-click interfaces and Fitt's Law (13). This made it easy for the user to enter or alter knowledge at the very moment of thinking it, and thus the process of expressing knowledge in new ways became much easier.

Istar was therefore thought to be a good starting point for the ICT project. The technical aim of the project was to investigate, through action research, what features are useful in conceptualizing such knowledge-poor domains and what steps and approaches are useful in knowledge acquisition. We report the findings below.

3.3 Knowledge Base Construction

Knowledge concerning trust is scattered in various forms, including formal documents, books, expert opinion, accepted norms and usages, etc. In our case the main source of knowledge came from a framework document for CPSs (4), supplemented by intense discussions with four partners with expertise in communications security, but who are nevertheless 'journs' as far as trust is concerned. In addition, a small number of international experts were interviewed using a questionnaire devised for the purpose.

Since trustability is the main criterion in the examination of a CPS, we have two goals: 'Can trust' and 'Can't trust', both Bayesian, whose values are interpreted as indicating the presence of reasons for trusting and distrusting the authenticity of the sending entity. The leaf nodes represent entries in the CA's CPS. Experience has shown that while the absence of some CPS entries or their lax appraisal might signify a lessening of trust and therefore contribute to the 'Can't trust' goal, some safety-related entries would contribute to 'Can trust' though their absence would not necessarily increase 'Can't trust'. As far as possible, the two goals are treated as independent from the point of view of Bayesian inference, and combining them for a final result is carried out by the interpretation of the user. A version of the inference net is shown in Fig. 2.

Fig. 2. ICT Inference Net

The result of the assessment of a CA is thus in the form of two scores, each reflecting the degree of affirmation in the two goals,. The final judgement, as to whether or not to accept the findings of this particular CA in the current commercial circumstances, is left to the relying party. If there are reasons for both trusting and distrusting, then such reasons are sought by examining the values of other, intermediate, nodes. One advantage of using an inference net is that the syntax of the net matches closely the semantics of the knowledge, so that each node often represents some significant factor in the knowledge domain.

4. PROJECT FINDINGS

The findings of this project are divided into two areas, the first concerned with the knowledge and the acquisition thereof, and the second concerned with software features that facilitate working in knowledge-poor domains.

4.1 Findings Concerning Knowledge and its Acquisition

Effect of Knowledge Source

The type of knowledge source can have a marked effect on the structure of the KB. Working from a hierarchical document like the CPS framework tends to produce a tree-like inference net with only a small number of multiple consequents (Fig. 2). This is because the knowledge is contained within a tightly structured arrangement with a nested hierarchy of sections, paragraphs and subparagraphs. This is unlike the situation in which the knowledge is being elicited from a human expert through interrogation, where one can expect a larger number of cross-linkages and counter argument. The reason is that such documents are often a deliberate simplification of actual expertise for purposes of clarity, memorability or authority. Such simplified expertise alone cannot yield a high quality KB, so it must be 'cultivated' to deeper levels (7) by the knowledge engineer proactively seeking its enrichment through discussion with experts, even when they only 'journs'.

Development of Knowledge Engineer

The knowledge engineer is the recipient of information from a wide variety of knowledge sources, and thus has considerable responsibility for making appropriate interpretations. At the start of the project s/he will usually have only a naïve view of the domain, but in established domains the experts can usually guide such interpretations and gradually the knowledge engineer develops considerable expertise of their own. However, in a domain where expertise has yet to become established this guidance is less effective, and the 'journs' are themselves developing expertise on the fly. Therefore arriving at the most appropriate conceptualization of the domain can take longer - and such conceptualizations often seem obvious in retrospect. An example of this from the ICT project was: the various parties involved were initially lumped together as mere "users" and it took time to recognise that they will have quite different concerns and interests and thus had to be distinguished into relying parties and sending entities.

However, while we may feel happier with a non-tree-like inference topology, closer to the imagined structure of multifacetted human inference, we should be careful of the introduction of the knowledge engineer's views. There are two issues here. One is justification of the knowledge, for instance in a court of law; strict adherence to the knowledge contained in the published work, even though simplified, is easier to justify than a person's interpretation. The other is that discussion with human experts might overly concentrate on particular examples and special cases and thus obscure the true pattern of inference. The broader picture of general inference relationships should be continually brought back into focus.

Structure of the KB

We found the structure of the original inference net was improved by the introduction of intermediate subgoals. This was achieved by taking one of the leafs (framework entry) linked to a goal and asking "Why?" For instance (4) has an entry concerning uniqueness of names, which is deemed relevant to trust. So we ask "Why?" and

discover that if two entities with identical trademarks e.g. Apollo, were allowed to use this name as their (non-unique) identifier in a certificate, then genuine confusion between the two certficates could arise, which could lead to either the possibility of sending confidential information to the wrong party. Thus trust in authenticity is lessened.

We also found a "What else?" question useful, both for determining extra antecedents not mentioned by Chokhani and Ford, and also to discern alternate inference paths to the goal. For instance, non-unique names can also make it easier for parties to masquerade as others and, again, trust is lessened.

In this way, using these two questions, which are two of the four questions mentioned by (14), the knowledge from the CPS framework was 'cultivated' (7) and given a more semantic feel. Confidence would be gained by the builder that real understanding was being increased. However, as seen from an interim KB shown in Fig. 2, this process is not complete.

In more complex situations involving choice, it was tempting to set up an enumeration data type to itemise the alternatives available. But it was found that, at least in the initial stages, a close mapping between the entry and such a type was not helpful, because it tended to obscure the inferential patterns being elicited. In the initial stages specific details of type should not be of too much concern. In any case, the particular type of construct to model a certain feature can easily be changed at a later stage, if necessary.

4.2 Findings Concerning KBS Software Features

Since the process of creating knowledge bases in knowledge-poor domains differs from that of assembling knowledge bases in well established domains, we can expect a different set of features in KBS software to be important. Here we identify and discuss what features have been particularly important in the ICT project thus far, not as an authoritative list, but rather to initiate debate.

Surprisingly, few attempts have been made to identify what software features aid knowledge engineering, beyond general statements of the need for such things as expressive power and ease of use. In the proceedings of *Expert Systems '97* not a single paper out of the 39 published had an explicit discussion of this; only (15) moved anywhere near this, listing some features added at the request of the user. This is a notable lack since KBS software should contain features tailored to the process of knowledge engineering as opposed to programming, and it should not be left entirely to the commercial software developers to determine what these features are. It is for this reason that we present an initial, and very incomplete, list of some features that we have found important in handling this kind of knowledge.

Visual Knowledge Representation

As expected, the visual nature of the knowledge representation language proved useful for two purposes, to ease construction and subsequent alteration of the KB, and to gain an holistic view thereof. Istar employs a very simple visual style, even omitting arrows on links and avoiding the display of too much information. As discussed in (16) important information came through tacit conventions like left-to-right inference and through the pattern of links around a box rather than in the explicit symbols. However, one symbolic effect that has proved important in this project was to distinguish visually between links that have a positive effect from those that have a negative one.

The importance of such 'visual cues' in Istar's user interface is one of the issues discussed in (16). Another is the mimimization of cognitive load and the avoidance of interruption of the knowledge engineer's thinking process by carefully 'grading' the load imposed by each operation. We found Istar's user interface appropriate here, though it had been designed for a different type of knowledge generation.

Separate Texts

Because of the importance of human interpretation in ill structured domains of knowledge, Istar offers at least four different texts for each attribute: a label (which is short enough to be displayed in the box representing the node), a meaning (which is a sentence of arbitrary length intended to record precise meanings, and which is shown in a window when the mouse moves over nodes), a question text (which is displayed if the node is put to the user as a question) and explanation text (which augments the question text). The label and meaning texts are shown during KB development and the question and explanation texts during runs with the user. If any text is missing then the next in a defined order of priority is used.

We found that use was made of all four texts, for different purposes. As in the earlier INCA project, the meaning text was important both to force precision in discerning the meaning of each node and also as a record for later use.

We found however that several explanation texts would have been useful during the run because there were several things the user wished to know:

Reference to source document (where appropriate) # Brief statement of what the thing in the question referred to, including its context # Brief statement of what its significance was for the goal # An example.

We found that many of these had to be provided in discussion with experts.

Further, there should perhaps be two versions of some of these texts, one for knowledge-poor experts and one for ignorant users. How to expand Istar to accommodate these is being considered. Cawsey (17) discusses a variety of explanations, differentiating them along three dimensions of role, content and type.

Seeing the Consequence of Questions

In a large, complex inference net, it has proved useful to display parts of the inference net, leaving the remainder hidden, especially the network of antecedents or consequents of a given node. Because for Istar inference takes the form of graph search rather than rule-firing, it has proved relatively easy to implement such facilities, and a second button was added to the user question panel which, when hit, displayed the entire consequent net of the current question. In this way, a display was given of the significance of the current question node in terms of what other nodes it influences, whether directly or indirectly. This feature was particularly impressive during demonstrations.

Exploring the Result

In addition to giving a visual representation to inference, the Istar easel can also be thought of as a guide to giving a reasoned analysis to a result: "Why has this particular

goal value resulted in this case?" This type of question can be answered, at least partially, by Istar's facility to show the antecedent net of any node, backtracking along the inference path which led to it and displaying what it finds. However, the facility needs to be made more sophisticated, to indicate the degree and direction that each leaf has on the value of the goal and hence on the degree of trust.

Modifying the KB During Run

During development, four main kinds of modifications to the KBS were encountered: to types and weights, to topological structure, to texts and to the sequence in which questions were put to the user. Few changes to numeric weights were made during the early stages of development, while the structure of the KB was being developed.

Many changes were motivated by demonstrating the KBS inference session to 'tame' experts. We found that what raised most discussion was not the results produced so much as the wording of the questions and of explanation texts. This agrees with the highly interpretive nature of the meaning of trust, in that the running of the KB stimulated the emergence of a variety of interpretations of each question as it appeared. Such discussion led, in the main, to changing of question and explanation texts, less commonly, to label and meaning texts, and occasionally to altering the local structure of the inference net or the type or weight of an node.

We found that it was essential to be able to make the alterations immediately they were suggested, so that ideas are not forgotten or distorted. To do this, without aborting the run, requires the KBS software to be multi-threaded in nature and robust against changes in KB structure. Istar possessed both of these qualities, so that a button could be added to the user question panel to allow access to the KB details and structure, and this new feature was frequently made use of.

Inverting the Meaning

An operation we found was commonly required was to invert the meaning of a node and therefore, with it, of each of its links, as shown in Fig. 3. This was often because the original meaning had been a double negative, which could confuse, so, during discussion, it was decided to reverse its meaning. This involves not only changing the various texts, but also modifying each arc, antecedent and consequent, of the node. At the symbol level at which most KBS software operates this is a cumbersome process and error-prone, but at the knowledge level it is an atomic operation. Therefore it deserves to be made a simple, single-button, action in KBS software.

|-|IG "r:kgref/ES98 Fig 3" -w5.55 -h4.5 -c -ra

|-|CE Fig. 3. Inverting the Meaning of a Node

5. CONCLUSIONS

Paul (7) and others have discussed 'knowledge-poor' domains, in which there are no true experts because real expertise has yet to build up. Creating a knowledge base for such domains takes the form of knowledge generation, rather than mere acquisition. This paper identifies four types of knowledge generation. In most of these, the very nature of knowledge engineering changes from what Winograd (18) calls a 'constructor's-eye-view' to a 'designer's-eye-view', and this has significant implications for both techniques and KBS software.

This paper generalises the findings from an attempt to build a knowledge base to calculate trustworthiness of the authenticity of Internet information. The project demonstrated some of the characteristics of knowledge-poor domains and the need for techniques tailored to knowledge generation. Not only is there no established body of expertise, but Internet trust is a highly interpretive notion, very context dependent and volatile in its meaning. The knowledge engineer might work from documents, but finds these simplified, of might engage experts in discussion, but finds they are only 'journs' as far as this issue is concerned. So to construct a good knowledge base needs special techniques.

KB construction is no longer a linear sequential activity, but more of a cyclical, continuous process in which the KBS software must become 'proximally' (3) part of the knowledge engineer. The style of user interface is important, and has been discussed in (16). By generalising from the experience with Internet trust, this paper discusses other features that are important, such as visual representations, the variety of explanations needed or the ability to modify anything in the KB as soon as the participants think about it. To modify things during a run demands a special robustness of the software!

ACKNOWLEDGEMENTS

REFERENCES

1. COLLINS H.M. The TEA-Set: Tacit knowledge and scientific networks, Science Studies, 1974, v.4, pp. 165-186.
2. BASDEN A., HIBBERD P.R. User interface issues raised by knowledge refinement, International Journal of Human Computer Studies, 1996, v.45, pp.135-155.
3. POLANYI M. The Tacit Dimension, 1967, (Routledge and Kegan Paul).
4. CHOKHANI S., FORD W. Internet Public Key Infrastructure, Part IV: Certificate Policy and Certification Practices Framework, an Internet-draft of the Internet Engineering Task force, July 1997.
5. BASDEN, A. On the application of Expert Systems, International Journal of Man- Machine Studies, 1983, v.19, pp.461-477.
6. HINES J.G., BASDEN A. Experience with the use of computers to handle corrosion knowledge, British Corrosion Journal, 1986, Vol.21, n.3, pp.151-156.
7. PAUL J. Building expert systems for knowledge-poor domains, in Bramer M.A., Macintosh A.I. (eds.), Research and Development in Expert Systems, X, 1993, (IEE BHR Group, London), pp.223-234.
8. EDEN, C. Perish the Thought, Journal of the Operational Research Society, 1985, v.36(9), pp.809-819.
9. MOORE E.A., AGOGINO A.M. INFORM: an architecture for expert-directed knowledge acquisition, International Journal of Man-Machine Studies, 1987, Vol.26, pp.213-230.
10. EDEN C. Using cognitive mapping for strategic options development and analysis (SODA), in Rosenhead J. (ed.), Rational analysis for a problematic world: problem

structuring methods for complexity, uncertainty and conflict, 1989, (John Wiley, Chichester, UK).

11. BASDEN, A., BROWN, A.J. Istar - a tool for creative design of knowledge bases,Expert Systems, 1996, v.13(4), pp.259-276.

12. NORMAN, D.A. Why interfaces don't work, in Laurel B (ed.),The Art of Human-Computer Interface Design, 1990, (Addison-Wesley), pp.209-219.

13. CARD S.K., MORAN T.P., NEWELL A.The Psychology of Human-Computer Interaction, 1983, (Lawrence Erlbaum Associates, Hillsdale, NJ, USA).

14. BASDEN, A., WATSON, I.D., BRANDON, P.S.,Client Centred: an approach to developing knowledge based systems, 1995, (Council for the Central Laboratory of the Research Councils, UK).

15. SUGDEN, R.C., HUME, S.J., Developing a common knowledge authoring tool for heterogeneous target systems in the PRODIGY project, in Macintosh A, Milne R (eds.),Applications and Innovations in Expert Systems V, 1997, (SGES Press, ISBN 1 899621 19 9), pp.71-82.

16. BASDEN A., BROWN A.J., TETLOW S.D.A., HIBBERD P.R. The design of a user interface for a knowledge generation tool,International Journal of Human Computer Studies, 1996, v.45, pp.157-183.

17. CAWSEY, A.Explanation and Interaction: The Computer Generation of Explanatory Dialogues, 1992, (Bradford Books, MIT Press, London, UK).

18. WINOGRAD, T. From programming environments to environments for designing,Communications of the ACM, 1995, v.38(6), pp.65-74.