

Case-Based Reasoning Systems for Knowledge Mediation

A.D. Griffiths, M.D. Harrison and A.M. Dearden

Human-Computer Interaction Group, Department of Computer Science,
University of York, Heslington, York YO10 5DD,
United Kingdom

Tony.Griffiths *or* Michael.Harrison @ cs.york.ac.uk

ABSTRACT: A knowledge mediation system eases the transfer of knowledge between two communities of practice. We describe one such system, based upon case-based reasoning, which reuses cases representing previous instances of mediation between two different communities. The system's specific purpose is to support management processes to plan the maintenance and operation of a utility distribution network, by mediating knowledge from the conceptual world of engineers in the business. This paper focuses on two particular issues. The first is the means by which a case is represented so that it characterises an instance of translation between the two conceptual worlds and the second concerns the mechanisms in the system for adapting the previous case to the current problem.

KEYWORDS: Knowledge mediation, case-based reasoning, communities of practice.

1. INTRODUCTION

Recent literature in Computer Supported Co-operative Work has drawn attention to problems that arise in the interface between so-called communities of practice (Wenger, 1998). In order to carry out their work efficiently, communities of practice typically make use of a specialised vocabulary, which eases communication about the work that has to be carried out. In addition, repetition and the internalisation of particular forms of work mean that their activity may be based substantially on implicit and tacit knowledge. However, specialised vocabularies are not necessarily understood by adjacent communities and differences in language and knowledge mean that the *boundary objects* (Star, 1995) which provide potential interfaces between the two communities can be interpreted quite differently from the two perspectives. A process of *knowledge mediation* is therefore needed when knowledge and information is moved between the two communities to provide the

translations and *transformations* that will allow the communication to be assimilated by the receiving party.

This paper is concerned with the proposal that software systems, if correctly designed, can make a contribution to *improved* knowledge mediation between different communities of practice. We describe a specific development, which is designed to mediate knowledge between two such communities in response to the need for detailed and rapid interactions between the two groups of workers.

Although the system has not been completed we have prototyped parts of it. These parts will be described in the current paper, along with the relevant design issues. In a future paper we shall describe the evaluation of the system. The development is intended to explore generic issues of technology and knowledge mediation. Its *specific* purpose, however, is to support business planning carried out by company managers. We have taken, as a case study,

the task of business planning undertaken by Northern Electric Distribution Limited (NEDL or ‘the business’). NEDL is an asset management company within the recently privatised Northern Electric group. NEDL owns and operates a large network distributing electricity to consumers in the Northern Electric area. ‘Business planning’ in this context means deciding how money will be spent on maintaining and operating the distribution network. This is a difficult task; one engineer told us that the situation of the business is characterised by “... increasing expectations, falling revenue and an ageing system”. Pressure is placed on the business by the scrutiny of an external regulator appointed by the government. The regulator has power to penalise the company if very high standards of reliability are not met while at the same time cutting the charges, which may be made by the business for the transport of the utility. The business must consider all its activities carefully!

Business planning is an uncertain process. However, some of the uncertainty *currently* associated with the process could be resolved by information which is already ‘known’ or recorded within the organisation, but requires substantial transformation or interpretation to answer business planning questions in forms that are posed by business managers. In particular, the engineering teams who make up the majority of the headquarters’ staff of the business are routinely involved in creating or accessing a number of rich sources of information. These include the current maintenance policy of the business, the asset registry, including information about the age and condition profile of the components of the distribution network, and fault records detailing the incidence of failures in different parts of the network. All of this information is potentially very relevant to issues arising in business planning, but is not accessible directly by the business planners. It is only currently accessible through interaction and negotiation with the engineering personnel since the knowledge, tacit or otherwise, held by the engineers about the existence, location and structure of this information and about how its representations should be interpreted, is not necessarily shared by the business planners.

The motivating concept behind our ‘knowledge mediation tool’ is therefore to *increase the value of information sources already available* within the business by decreasing the cost of exploiting this information within the community of practice of the business managers. We argue that technology supporting knowledge mediation between the conceptual world of the business planners and the conceptual world of the engineering teams might

decrease the cost to the business planners of accessing some of this information, reducing *some* of the uncertainty about future outcomes associated with business planning through the provision of extra information. In addition, by reducing the cost to the business managers of obtaining information in relevant forms, the scope of the analytical investigations that might be carried out in the course of business planning is broadened.

The remainder of the paper discusses the design of a prototype ‘knowledge mediation’ system, designed in response to these issues. Section 2 outlines the system and explains the role played by Case Based Reasoning. Section 3 describes the notation used to describe cases so that they are instances of translation between conceptual worlds. Section 4 describes the means by which the system supports the adaptation of a retrieved case to the current problem.

2. A ‘KNOWLEDGE MEDIATION SYSTEM’

The designated role of our prototype system is to facilitate knowledge mediation so that information, which previously was only accessible to the business managers by interaction with the engineering community, is made available at a decreased cost. Fortunately, most of this information is stored in the form of relational databases on electronic media. Other relevant knowledge, which is currently only represented on paper, could also be coerced into this form. The knowledge mediation system will stand as a proxy for the engineers who currently access these information sources, allowing the business managers greater independent access to the information. To achieve this, the system must do two things:

- It must provide a user interface where the business manager may express requirements for business planning information and be presented with this information appropriately aggregated in an accessible form.
- It must encode the engineer’s knowledge of the databases and provide the processes, which will translate and transform the information into forms that are digestible by the business manager.

In our system, the former is provided by a spreadsheet style interface where the business manager can create schematic representations of the required information, while the latter is provided by the knowledge-based technology of *case-based reasoning*. These two aspects of the system are explored in the following sub-sections.

2.1 User Interface to the Knowledge Mediation System

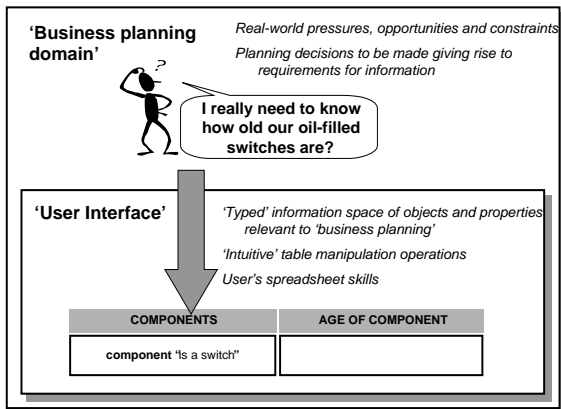


Figure 1: The user interface to the knowledge mediation system

Figure 1 illustrates the requirements for the user interface to the knowledge mediation system. The business manager must be given the means to translate requirements for information arising during the business planning process into some kind of formal representation of those requirements. For this purpose, we are developing a graphical representation for the conceptual structure of tables. We assume that the idiom and usage of the spreadsheet package are familiar to business managers (a legitimate part of the business planning domain) and in this way we have attempted to minimise the additional skills that must be assimilated by the user of the knowledge mediation system. Associated with the table representation are a set of 'intuitive' table manipulation operations by which requirements for information can be expressed. These operations are:

(1) Specifying a **type** which indicates *what kind of object* is described by each row of the table. In general, the types in the script language must be recognisable categories of object from the point of view of the business manager. Types playing a prominent role in our prototype system include **assets** of the business, sub-types of asset such as network **components** or operational **locations**, or associated entities such as fault incidents or maintenance tasks. In addition, the user may specify combinations of types in a single table.

(2) Adding **data requirements** to the table. These indicate what is wished to be known about the entities which are represented by the records (rows) of the table. They are *properties* of the different types of object known to the system and are shown in the graphical representation as additional columns added to the table representation. In Figure 1, the 'Age of Component' has been chosen as a data requirement.

(3) Adding **conditions** to the table. These determine the choice of rows (records) that are to be included in the table. The table represented in Figure 1 will contain only components of a certain subtype ('switches').

(4) **Aggregating** tables. The user can specify aggregations of tables in order to create a digestible summary of information, for example summing items of expenditure according to geographical location or using budget headings showing the type of expenditure.

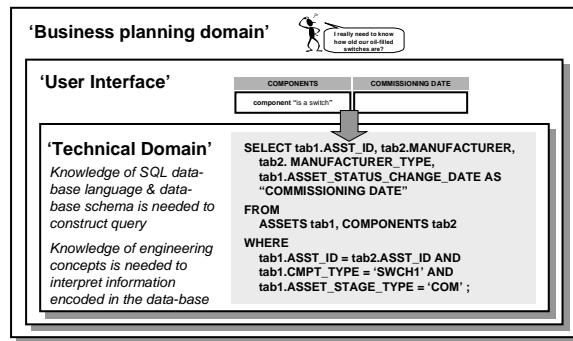


Figure 2: Internal functions of the knowledge mediation system

Figure 2 illustrates the requirements for the internal processing functions to be provided by the knowledge mediation system. The schematic representation of 'information required' must be translated into an executable database query in order to populate the business model created by the business manager. Many different kinds of knowledge are needed to carry out this translation and to transform the retrieved data into the form required by the user. The following kinds of knowledge, needed to use a database system and presumably part of the tacit knowledge of the engineering community, can be differentiated:

(1) *Schematic Knowledge*: knowledge about what different kinds of information are available and where this information can be accessed; detailed knowledge about the way in which the represented information has been organised. In the case of relational databases, this includes knowledge of the way in which the information is divided up into relations and the names of the attributes defining each relation.

(2) *Procedural Knowledge*: procedural knowledge and skills necessary for accessing the information. In the case of relational databases, this includes knowledge of the SQL query language through which the databases are accessed.

(3) *Usage Knowledge*: knowledge about the way that the database has been populated in practice; knowledge about the completeness of different

representations of information and about the reliability of any information recorded.

(4) *Interpretative Knowledge*: domain knowledge about the 'meaning' of any represented information, and particularly knowledge about how this can be translated into knowledge at the 'business level'.

We have chosen to take a 'case-based' approach to the representation and manipulation of this knowledge. This choice is explained in more detail in the next sub-section.

2.2 Case-Based Reasoning in the Knowledge Mediation System

Case-based reasoning (CBR) systems (Dearden & Harrison, 1997) are knowledge-based systems which use a memory of relevant 'past' cases to interpret or solve a new problem situation. "Rather than creating a solution from scratch, a reasoner using case-based reasoning recalls cases *similar* to its current problem situation and solves or interprets a problem by reasoning with past solutions and interpretations" (Rissland and others, 1989). Case-based reasoning offers a number of advantages for building software systems in large and complex domains such as the one we have described. In particular, the knowledge content of the system can be built up incrementally, as the system is used. This may be achieved by simply adding new cases to the case memory as they are created in the course of problem solving. As a result of this process, the knowledge of the system will be most comprehensive in precisely those areas where the system finds most use.

In order for case-based reasoning to be used successfully, there must be a stable correspondence between problem situations and solutions to those situations. A solution, which is a sufficient response to a particular situation on one day, must suffice again on a different day if the same problem recurs. In addition, it must be possible to identify when a new problem situation is different but *analogous*, or sufficiently *similar*, to a previously occurring problem for which a solution is known. In a rich and changing environment such as business planning, it may not be clear that these conditions can be met. However, we felt justified in assuming a number of characteristics of interactions between business managers and engineers, after a careful investigation of the work environment. The first assumption is that there is sufficient stability about work in the engineering domain. The sources of information will remain sufficiently static that it makes sense to consider the reuse of previous instances of access to these information sources. The second assumption is that, although business-planning scenarios might vary

widely and appear to be unique to the current situation, the procedures by which these scenarios are analysed may have a significant degree of commonality. With these assumptions it makes sense to explore the role of an Interactive Case Memory (a form of CBR system where precedent cases are retrieved and re-used in co-operation with a user of the software system) as a basic technology for the Knowledge Mediation System.

The case-based knowledge mediation tool will therefore provide a record of previous examples of business planning episodes where the "engineering community" provided information in response to requests by the business managers. This memory will be in the form of sets of tables of information of the kind manipulated in the user interface of the knowledge mediation system along with the executable database queries necessary to populate the tables. The interface provides access to these previous episodes in terms that are meaningful to business managers and supports *adaptation* of the "nearest" example so that it can be used to inform the scenario currently being evaluated.

Several research issues are important here not only to the CBR community but also from a human computer interaction point of view:

(1) How to represent the cases in a way that supports the demanding requirements just stated. The 'script language' that we have developed for this purpose is described in Section 3, below.

(2) How to re-use knowledge represented in this language in the context of a new business planning situation. One particular problem is that some of the information represented in our cases, i.e. the database queries associated with each table, is not comprehensible to the user. If the most similar case in the case memory is only a *partial* match to the current problem situation, then the user of the system will be unable to adapt these parts of the case and, instead, automated adaptation must be carried out by the system. These mechanisms of adaptation are discussed in Section 4, below.

(3) Whether it is possible to populate a case-base so that the processes of matching and adaptation described in Section 4, below, are feasible. These issues will be considered as part of the future evaluation of our prototype system.

3. REPRESENTATION OF MEDIATED KNOWLEDGE

We have developed a 'script language' to represent *instances of translation* between the conceptual world of the business manager and the conceptual

world of the engineering teams. In terms of knowledge mediation, three main requirements have applied to the development of this language:

- (1) Each case contains a table or set of tables representing the information required by the business manager and its desired presentation.
- (2) Each case must include, in addition, the executable database queries that are needed to populate the table specified by the business manager.
- (3) Each case must be represented in a way which facilitates the adaptation of retrieved cases to meet the needs of a new problem situation.

Existing notations for representing tables, relations and database queries are insufficient for a number of reasons. With respect to requirements (1) and (2) above, the script language plays a *mediating* role in bridging two different levels of representation. One is the *conceptual* structure of the table, as presented to the business manager in the user interface of the system, and the second is the *implementational* structure of the table, where the table structure is described by database invocations which must be provided by the engineering community of practice. At the first level, the elements of the table structure, specifically the *conditions* and the *data requirements*, are recognisable concepts in the business-planning domain and are named in terms chosen by the business manager. At the second level, these ‘user level’ concepts must be implemented in terms of the actual values and parameters which are recorded in the mediated information sources. This dual structure is not reflected in current table representation languages, which maintain only a single level of representation.

With respect to requirement (3), adaptation of existing table representations is difficult because conventional table description languages fail to express any *rationale*. For example, these languages will list the selection criteria which must be satisfied by any record that is to be included in the table, but do not show *why* each condition has been included or differentiate the roles played by the different conditions. In addition, there may be dependencies between conditions, which mean that it does not make sense to delete one of the conditions without deleting a number of them. Again, these dependencies are not normally represented. Without the explicit representation of the conceptual structure of the table, adaptation of existing table representations by any means, whether manual or algorithmic, is difficult.

Our approach, therefore, has been to develop a representation language which provides a restricted but sufficient sub-set of the manipulations possible in existing table manipulation languages. This language describes the table both at the conceptual (user) level

and the implementational (internal) level. It also organises table descriptions into meaningful chunks, where elements of the implementation level, which share in translating a user-level concept, are grouped together in a single syntactic unit.

Each table is represented in the script language by a sentence of the following form:

```
TABLE( NAME( TABLE_NAME ),
      OF( TYPE ),
      CUSTOMISERS( CUSTOMISERS ),
      PRESENT( PRESENTATION ) )
```

TABLE_NAME simply names this table for reference by other table descriptions. The *TYPE*, *CUSTOMISERS* and *PRESENT* keywords are interpreted *cumulatively* to reconstruct the represented table.

Firstly, the *TYPE* expression records the type assigned to the table by the system user (see Section 2.1, above). In particular, the script language is designed to represent *combinations* of types. For example, each row of a table might contain *asset* characteristics along with details of a *maintenance task* associated with that asset.

Secondly, the *CUSTOMISERS* clause modifies the ‘base-table’ defined by the *TYPE* expression. A ‘customiser’ is either an additional data requirement or a condition added to the definition of the table.

Thirdly, the *PRESENTATION* of a table provides the rest of the information needed to display the table. This includes; the name given to each of the columns of the table; the order in which the columns of the table will be presented to the user; the aggregation of the rows of the table. An ‘aggregation attribute’ associated with each column of the table specifies the aggregation of the table. An aggregation attribute is either the name of an aggregation operator (COUNT, SUM, MIN, MAX etc) or the term NONE which indicates that the column is an aggregating column.

The script language representation of the problem-solving episode in Figure 1 is shown below:

```
TABLE(
  NAME( "Age of switches" ),
  OF( tp( components, [ ] ) ),
  CUSTOMISERS(
    DEF( cond, "is a switch", components,
        CMPT_TYPE = "SWCH", [ ] ),
    DEF( data, "Age of Asset", assets,
        ASSET_STAGE_TYPE = COM,
        global :: CURRENT_DATE -
        ASSET_STATUS_CHANGE_DATE ),
  PRESENT( [
    ( "ASSET#", NONE, ASST_ID ),
    ( "COMPONENT DESCRIPTION", NONE ),
    ( "AGE OF COMPONENT", NONE ) ] ) )
```

This example illustrates the mediating role played by the script language. A condition such as “is a switch” is a concept from the business planning domain and is displayed and manipulated in the user interface of the system. The script sentence shown above links this concept to its translation in the language of the data-base schema: `CMPT_TYPE = "SWCH"`. Similarly, the data requirement “Age of Asset” is associated with the condition and formula which translate this element of the ‘conceptual structure’ of the table into a piece of executable data-base query. ‘Real world’ translations may obviously be much more complex.

The similarity between representations of conditions and data requirements is deliberate, since we found that, although the distinction between the two is meaningful at the user level, in practice the two may require both a logical condition and a number of additional columns to be added to the data-base query. We asked an informant to demonstrate how items of plant, which had been replaced during the course of a recent asset replacement programme, could be identified in the asset database. In the course of this, the informant proceeded to translate the ‘commissioning date’ of a piece of plant (the date on which the asset entered service). The informant indicated the data-base attribute which recorded this information but simultaneously added the condition “`ASSET_STAGE_TYPE = 'COM'`” to the selection criteria in the query.

4. RE-USING MEDIATED KNOWLEDGE

Case-Based Reasoning systems are based around a characteristic series of processing steps e.g. (Aamodt and Plaza, 1994), where, following the presentation of a description of the current problem situation, stored cases which are potentially relevant to the new problem are *retrieved* from the case memory. These are inspected to determine which are most *similar* to the current situation; the selected cases are *adapted* to create solutions for the current situation and so on. Many variants of each of these steps have been explored according to the representational form given to the stored cases. Our task was to develop a case-based reasoning cycle, which could exploit cases encoded in the script language described in the previous section.

4.1 Retrieval of Stored Cases

Methods for the efficient retrieval of cases in a structured representation such as ours have been discussed in (Plaza and others, 1996; Tammer and

others, 1996). In these methods, an incomplete fragment of case is presented to the case memory, which returns the most similar, complete, stored case. This style of operation fits the requirements of knowledge mediation, since cases expressed in the script language of section 3 combine information from both the business planning viewpoint (the ‘conceptual’ structure) and the engineering viewpoint (the ‘implementational’ structure). However, the business planner using the system is only required to describe the information required on the ‘conceptual’ level, i.e. from the business planning point of view; the role of the case memory system is to provide relevant complete cases from which the missing ‘implementational’ level can be inferred.

The partial case representation, which will act as a query to the case memory system, is created through the interactions described in section 2.1, above. These operations create a graphical ‘table schema’ representing the information required by the user (Figure 1, above) and an *internal* representation of the same information in a sentence of the script language. Section 3 emphasised how the script language associates the ‘name’ of each element of a table specification with its ‘translation’. However, only the name of the element is provided by the user when a query is being created. These ‘names’ are chosen by the user from a list presented by the knowledge mediation system of elements which have already been defined in the cases held in the case memory. If the user believes that the concept being introduced into the query is genuinely new, then a new ‘name’ may be added. In either case, the system places a special ‘null’ term into the script language term to mark the location of the missing translation. Thus the user only provides those parts of the case representation corresponding to the conceptual structure of the table. Cases are retrieved by matching this partial case fragment to the more complete cases stored in the case memory.

The case memory and its associated retrieval mechanisms are currently being implemented.

4.2 Structure Matching & Adaptation

The knowledge mediation system must now compute a single representation in the script language which combines all the relevant information from the script representing the user’s query and from the scripts retrieved as cases from the case memory. We call the process that carries out this transfer of knowledge ‘structure matching and adaptation’. This process is based on algorithms described in (Jantke, 1994; Plaza, 1995) for transfer of knowledge between structured case representations. Taking a similar

approach, our process proceeds in two steps:

Structure matching. Taking the ‘query’ script with each of the retrieved cases in turn, the system calculates the ‘intersection’ of the two representations by ‘matching’ one representation onto the other. This results in a new sentence in the script language that represents precisely what the two scripts ‘have in common’ (i.e. their *similarity*). The process also generates two series of operators, which specify a series of transformations by which the two original scripts could be recovered from their intersection. These two series of operators thereby encode the *differences* between the query and the retrieved case.

Adaptation. The two initial scripts are set on one side, and the adaptation process takes as input their intersection and the two operator traces. The adaptation process traverses these two lists in turn, using a set of heuristics to decide whether the operation to recover the query script or the retrieved script should be used. The result is a hybrid of the query and the retrieved case. In particular, the adaptation process is required to replace any ‘null’ terms representing missing translations of user concepts in the query script with translations of those concepts derived from the retrieved cases. However, our implementation in fact does more than just filling in missing translations - elements of the retrieved case representing the *context of use* of the translated concept, such as other elements which refer to that element or the associated aggregations, may also be transferred to the query script.

We have developed a set of heuristics for structure matching and adaptation in the script language described in section 3, and have implemented a ‘structure matching and adaptation engine’ based on these heuristics.

4.3 Manual Refinement of Cases

The automatic process, described above, for the adaptation of retrieved cases, is an *heuristic* one, as in fact are all case-based reasoning processes. A necessary final step therefore is that the user of the system must ensure that the information that has been retrieved by the knowledge mediation system through the execution of the computed script is in fact the information that was expected. We envisage that the use of the knowledge mediation system is an iterative and interactive process. The knowledge mediation system must support at least three further interactions:

Validation of the results of knowledge mediation. In case there is any doubt about the information that has been presented by the knowledge mediation system, the user must be able to inspect the formulae

into which each of the elements in his query has been translated. Here, there are issues about how to support the comprehension of user as the ‘knowledge mediation threshold’ is crossed, which we hope to explore in the remainder of the project.

Iterative refinement of the user's query. Even where there is no doubt about the translation of the user's query, we would expect the user of an interactive system such as this to refine the concept of what information is required iteratively through interaction with the data that is available. The structure matching and adaptation process uses the same representational space for the inputs and outputs of case-based reasoning. It is therefore straightforward to use the same facilities of the user interface as have been used to create the initial representation of the requirement for information to refine the user's query.

Incremental addition of mediation knowledge. A final possibility is that the system contains no suitable precedent for the translation of the user's requirement for information into an executable database query. In this case, the knowledge mediation system must provide facilities for a suitable translation to be provided by the engineering community of practice and this translation to be added to the case memory. In this way the knowledge base of the system is incrementally extended and future breakdowns can be avoided.

5. CONCLUSIONS

It has been argued for a number of years that knowledge-based systems should be designed to empower users to carry out their tasks (Woods and others, 1994; Fischer and others, 1991). This suggestion may be contrasted with the ‘expert system’ philosophy of knowledge-based systems which attempts to solve problems *on behalf of* rather than *in co-operation with* the user. Our version of a ‘knowledge mediation system’ has aimed from the start to support the task of business planning rather than to automate or constrain it. Our approach to this has been two-fold. On the one hand, we began our development by using methods of enquiry developed for general “non knowledge-based” software development, for example scenarios-based methods (Carroll, 1995) and contextual enquiry (Beyer and Holtzblatt, 1998). Through these methods we sought an understanding of the work to be supported so that we could envisage the ways in which the technology might impact the work system. On the other hand, we argue that our knowledge mediation tool has the potential to empower users. The recall of previous instances and comprehension of their connection with

the current problem instance in terms that are relevant to the business manager provides access to detailed knowledge about the current state of the business so that it can be directed to supporting decision making. The algorithms that support the reuse of these previous instances and their matching to the current problem provide an appropriate level of control of the information that is relevant to the current problem.

The main issues in implementing case-based reasoning processes as part of our prototype knowledge mediation tool have been to do with developing a suitable representation for the storage of cases in the case memory, and to find appropriate algorithms for the retrieval and re-use of these representations. In particular we have focussed on the script language, although we have also attempted to outline the ways in which stored cases are exploited by re-use. The main requirement on the script language is to be able to associate concepts which are meaningful to the business managers using the knowledge mediation system with the translation of those concepts into data-base invocations and formulae which will construct the required information from the available data sources. This requires structuring the language so that each syntactic element of the language corresponds to a meaningful conceptual element of the query. The SQL database language does not have this property; if an arbitrary sequence of tokens is deleted from an SQL query, the resulting string will almost certainly not be a well-formed query. Our script language is designed so that removing any of the constituent elements of a script will result in a new script which still has a valid interpretation, and in addition the chunk that has been removed is a separable and independent piece of knowledge.

Future evaluation of the system in the context of use should address a number of issues associated with knowledge based tools of this kind, for example: 1) Problems of 'false certainty' associated with automatic translation. 2) Problems of breakdowns where the system is not providing the appropriate level of knowledge. We have created a proxy for the engineer, but still need to enlist the engineer to fill in gaps in the knowledge of the system.

Acknowledgements: We acknowledge with gratitude the support of Northern Electric Distribution Ltd and the UK Engineering and Physical Sciences Research Council (Grant GR/K84752).

REFERENCES

Aamodt, A. and Plaza, E. (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial*

- Intelligence Communications*, 7(1), 1994.
- Beyer, H. & Holtzblatt, K. (1998) *Contextual Design*. Morgan Kaufmann.
- Carroll, J. M., editor (1995). *Scenario-Based Design: Envisioning Work and Technology in System Development*. John Wiley & Sons.
- Dearden, A.M. & Harrison, M.D. (1997) A Software Engineering Model for Case Memory Systems. *Computer Journal* 40(4) pp. 167-182.
- Fischer G., Lemke A.C., Mastaglio T. and Morch A.I. (1991) The role of critiquing in cooperative problem solving. *ACM Transactions on Information Systems*, 9(3):123-151, April 1991.
- Jantke, K.P. (1994) Nonstandard concepts of similarity in case-based reasoning, in Bock H-H., Lenski W., and Richter M. M. (eds.), *Information Systems and Data Analysis. Proc. 17th Annual Conference of the Gesellschaft fur Klassifikation, March 1993*. Springer-Verlag.
- Plaza, E. (1995) Cases as terms: A feature term approach to the structured representation of cases, in Veloso M. and Aamodt A. (eds.), *Case-Based Reasoning Research and Development: Proc. ICCBR-95*. LNAI vol. 1010, Springer Verlag.
- Plaza, E., Lòpez de Màntaras, R. and Armengol, E. (1996) On the Importance of Similitude: An Entropy-based Assessment, in Smith I. and Faltings B. (eds.), *Advances in Case-Based Reasoning: Proc. EWCBR-96*. LNAI vol. 1168, Springer Verlag.
- Rissland, E.L., Kolodner, T. & Waltz, D. (1989) Case-based reasoning from DARPA: Machine learning program plan, in K. Hammond, editor, *Proc. DARPA Case-Based Reasoning Workshop, May 1989*, pp. 1-13. Morgan Kaufman.
- Star, S.L. (1995) The Politics of Formal Representations: wizards, gurus and organizational complexity, in Star S.L. (ed.), *Ecologies of Knowledge, Work and Politics in Science and Technology*. SUNY Press.
- Tammer E.-Ch., Matuschek D., Jantke K.P., and Steinhöfel K. (1996) Learning Case Classification for Improving Case-Based Reasoning, in *Proceedings: 9. Fachgruppentreffen Maschinelles Lernen der GI Fachgruppe 1.1.3, Universität Chemnitz-Zwickau March 1996*, Chemnitzer Informatik- Bericht CSR-96-06, ISSN 0947-5125.
- Wenger, E. (1998) *Communities of Practice: learning, meaning and identity*. Cambridge University Press.
- Woods, D.D., Johannesen, L.J., Cook, R.I. & Sarter, N.B. (1994) Behind Human Error: cognitive systems, computers and hindsight. CSERIAC SOAR 94-01

