A hybrid-expert-system based tool for scheduling and decision support

F. Franek
Dept. of Computing & Software
McMaster University
Hamilton, Ontario
L8S 4L7 Canada
email: franek@mcmaster.ca

V.L. Rosicky
Terren Corporation
4711 Yonge Str.
Toronto, Ontario
M2N 6K8 Canada
email: lanny@sqlx.terren.com

I. Bruha
Dept. of Computing & Software
McMaster University
Hamilton, Ontario
L8S 4L7 Canada
email: bruha@church.cas.mcmaster.ca

INTRODUCTION

The system described in this report arose from the needs of Terren Corporation that is in the business of designing and developing commercial multi-platform client-server software systems with a back-end relational database for tour operators (travel agencies) capable of handling all aspects of booking and marketing of their products. In many fundamental aspects the scheduling of booking actions or marketing activities are very common among tour operators, but they differ in many details and implementation from agency to agency. Thus, Terren Corporation faced a problem of how to design a general system that, nevertheless, can easily be tailored by the end-users themselves to their individual specifications and needs.

The research the authors engaged in had as the goal an answer to this problem. The solution we are presenting here is a scheduling and decision support tool that is based on a hybrid expert system integrated with and within the back-end relational database.

The roots of the expert system methodology used in this research lie in the McESE (McMaster Expert System Environment) project of McMaster University (cf. (Franek 1989), (Franek and Bruha 1989), (Lentz 1990), (Jaffer 1990), (Chlobowski 1991), (deSouze 1992), (Knyf 1993)) carried out by the first and third authors, and the TESS (Terren Expert System Shell) research efforts carried out by the first and the second authors in Terren Corporation.

In the following we shall refer to the system providing the planning and scheduling of actions as TESS as well. The Terren Corporation software systems in which TESS has been embedded are systems code-named JANEWAY (on a commercial basis) and GREENWICH (on a research and development basis).

A BRIEF DESCRIPTON OF THE SYSTEM

The framework of the TESS solution relies on two sources of knowledge for the backward-chaining rule-based expert system making the time-based plans of actions and activities.

The first source is a set of relations in the back-end database and the data stored in them that represent, in a very diffuse way not very conducive for intelligent processing, the knowledge needed to make the decisions concerning the sequence of actions to be taken and their timing. This is common to all users of the system.

The second source is the knowledge particular to each individual user. It is stored in the form of SQL queries in a special relation in the back-end database. Later we shall explain why the form of SQL queries was chosen.

TESS system works in three distinct stages.

1. In the first stage, TESS extracts the relevant knowledge from the first source and builds a set of TESS rules from it. We shall refer to them as the *system-level rules.*
2. In the second stage, it selects the relevant rules from the second source, translates

them into TESS format and adds them to the rule set built so far. We shall refer to these rules as *user-level rules*.
3. The resulting set is the temporary knowledge base used for the decision making by the expert system that is the third stage of TESS.

The extraction and selection are controlled by certain parameters that are supplied to the first stage of TESS by JANEWAY or GREENWICH when they invoke TESS (e.g. what booking, what trip, booking date, departure date etc.). The reason for this is to have the expert system work with a knowledge base that is as lean as possible to facilitate a reasonable response time.

The second stage of TESS compiles the knowledge base as prepared by the first stage. The compilation stage has several roles, but two main roles are the syntax and to a limited degree semantic check of the knowledge base just created, and the compilation of the knowledge base for speedier processing by the inference engine of the expert system of the third stage. Since the knowledge base is created on-line in real time, the checking is necessary and indispensable.

The third stage is an expert system whose inference engine processes the knowledge base repeatedly over a simulated time line (basically day by day) in order to produce an activity plan. For example, for a booking of a tour, the plan would typically include actions such as when a deposit must be made, when a full payment must be received, when a vaccination certificate must be presented (if at all), when and how cancellation penalties may be applied and so on.

After the plan of actions has been produced, the temporary knowledge base and its compiled form are both purged from the system.

The user-level rules used to customize JANEWAY or GREENWICH to the particular needs of the user are stored in the relational database on the server, thus shielding the user from the complexities of the underlying TESS and any need to interact with it. TESS rules have a complex syntax (see below) and semantics in order to provide high expressiveness needed to apply it to different problems in different domains. The user-level rules have much more restricted syntax and semantics on purpose: to allow the end-users to tailor the system without needing a training in the use of TESS.

These rules have the form of SQL queries and SQL subqueries are their integral parts, thus allowing for rules that can be triggered and controled by actual data stored in the database without any need for the user to facilitate such database access explicitly (e.g.: *check whether the payment has been made, and if not, cancel the booking* - the rule must include how to check whether the payment has been made by querying the database using an SQL query and possibly by changing the booking status by modifying some values in the database using an SQL query). The additional advantage of storing the user-level rules in the form of SQL queries lies in the fact that both JANEWAY and GREENWICH provide a user-friendly GUI-type query builder and SQL queries are needed for other aspects of the system and thus can be used easily for maintenance of the user-level rules.

TESS RULES

The language of TESS-rules is rather general with a high level of expressiveness (this can be seen from the fact that such diverse constructs as neural nets and Petri nets can be simulated within the expert system using such rules (cf. (Franek and Bruha 1991), (Franek and Bruha 1995)). Unlike "pure" rule-based expert systems where the knowledge is represented in a declarative form, TESS introduces to some degree a procedural knowledge representation (see cvp functions below).

Before we can explain the previous statement exactly, we have to describe a general form of a TESS rule. Rather than providing a formal definition an example of a rule exhibiting all of possible features will illustrate it sufficiently.

```
r1:     (0.3*P(x,"abc")[>=.3] &
        ~Q(x,y)) |
        R(2,3.4,y)
 ==and:f1,or:f2=>
        0.7*~S(y,x)[>.6]
```

r1 is the ***rule identifier***. The left hand side of the rule is a Boolean combination of ***terms***, where a term consists of a ***weight*** (if omitted the default is 1, otherwise it must be a real literal whose value is between 0 and 1 inclusive), a ***predicate*** with list of ***arguments*** (the list may be empty), the arguments may be ***strings***, ***integers***, ***reals***, or ***variables***. The predicate may be negated. The predicate may be followed by a ***threshold directive***.

~ represents negation, **&** represents “and”, and | represents “or”. The threshold directive has the form **[op val]** where **op** is a relational operator (one of =, !=, <, <=,>,>=) and **val** is a numeric literal whose value is between 0 and 1 inclusive. The right hand side of the rule consists of a single term. The **arrow** `=and:f1,or:f2=>` defines which functions should be used to process “and” and “or” - if omitted, “and” is processed as a minimum and “or” as a maximum (i.e. the standard fuzzy logic interpretation of “and” and “or”). The functions to process “and” and “or” are collectively referred to as ***cvp functions*** (certainty value processing functions).

When the rule **r1** fires, first the left hand side expression is evaluated. The evaluation process consists of evaluation of individual terms. The value of a conjunction of two terms is processed by the cvp function prescribed for “and” (in our example by function **f1**, i.e. val(t1 & t2) = f1(val(t1),val(t2)), and the value of a disjunction of two terms is processed by the cvp function prescribed for “or” (in our example by function **f2**, i.e. val(1 | t2) = f2(val(t1),val(t2)). The value of a negated term is a complement in 1, i.e. val(~t)=1-val(t). The value of a term is obtained in the following manner: first the value of the predicate with concrete arguments (i.e. all variables must be instantiated) is obtained (how exactly will be described in the working of the inference engine below). If the term contains a threshold directive then the value of the predicate is further modified, otherwise it is not. The modification is based on whether the value of the predicate satisfies the threshold directive, if so, the value is modified to 1, otherwise reduced to 0. If a weight is specified, the resulting value is just multiplied by the weight. Once the value of the left hand side expression is known, it is assigned to the right hand side predicate (it again may be modified by negation, weight, or threshold directive).

CONCLUSION AND FUTURE RESEARCH

TESS rules and its inference engine are capable of various approaches to dealing with uncertainty (in essence extending the traditional Boolean truth values to any other range), among others fuzzy logic and use of neural nets (cf. (Jaffer 1990), (Franek and Bruha 1990), (Knyf 1993)). At the present, TESS within JANEWAY and GREENWICH is working strictly with classical Boolean values. In the area of booking actions, the Boolean approach is rather sufficient. In the area of marketing activities the situation is much more complex and less clear, and the domain expertise is fraud with vague and imprecise terms. The authors at present time are researching the two approaches mentioned above, fuzzy logic and neural nets, to provide the treatment of uncertainty inherent in decisions concerning marketing activities.

The SQL format of TESS rules and their intimate connection with the back-end relational database are currently being researched by Terren in the context of a possible enhancement of database functionality. TESS in conjunction with triggers can provide a natural way of incorporating another layer of "active rules" in the database (cf . (Ceri and Fraternali 1997), (Paton 1998)), insuring that the database can respond to events as they take place going way beyond the usual triggers, if it is desirable for the application. That research, though, is outside of the scope of this expository paper.

REFERENCES

S.Ceri and P. Fraternali, *Database Applications with Objects and Rules*, Addison-Wesley, 1997

A. Chlobowski, "McESE Extension of Allegro Common Lisp", M.Sc. thesis, McMaster University, 1991

F. Franek, "McESE-FranzLisp: McMaster Expert System Extension of FranzLisp", in *Computing and Information,* North-Holland, 1989, 379-382

F. Franek and I. Bruha, "McESE-McMaster Expert System Environment", in *Computing and Information*, North-Holland, 1989, 383-388

F. Franek and I. Bruha, "An environment for extending conventional programming languages to build expert system applications", *Proceedings of the 1989 Expert Systems Theory & Applications International Symposium* (Zurich, Switzerland, June 1989). IASTED, 106-109

F. Franek and I. Bruha, "A way to incorporate Neural Networks into Expert systems", *Proceedings of the 1990 Artificial Intelligence Application & Neural Networks International Symposium* (Zurich, Switzerland, June 1990). IASTED, 251-254

F. Franek and I. Bruha, "Simulation of Neural Nets in Expert System Environment McESE", Modelling and Simulation 1991, *Proceedings of the 1991 European Simulation Multiconference* (Copenhagen, Denmark, June 1991), SCS Europe, pp. 73-78

F. Franek and I. Bruha, "Simulation of Petri Nets in Rule-Based Expert System Shell McESE", *Proceedings of the 1995 European Modelling and Simulation Multiconference* (Prague, Czech Republic, June 1995), SCS Europe, 173-177

Z. Jaffer, "Treatments of Uncertainty and Their Emulation in McESE", M.Sc. thesis, McMaster University, 1990

D.L. Knyf, "Incorporation and training of neural networks in McESE expert systems", M.Sc. thesis, McMaster University, 1993

D.H. Lentz, "Notes on the implementation of McESE Scheme", Tech. report 90-01, Dept. of Comp. Sci. and Systems, McMaster University, 1990

S.M. de Souza, "Requirements Analysis and Software Design of the Extended McESE System", M.Sc. thesis, McMaster University, 1992