Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

SIMSELECT : A SYSTEM FOR SIMULATION SOFTWARE SELECTION

Vlatka Hlupic
Amardeep Singh Mann

Department of Computer Science and Information Systems at St John's
Brunel University
Uxbridge, Middlesex UB8 3PH, UNITED KINGDOM

ABSTRACT

SimSelect is a system developed for the purpose of providing support for users when selecting simulation software. SimSelect consists of a database which is linked to an interface developed using Visual Basic 3.0. The system queries a database and finds a simulation package suitable to the user, based on requirements which have been specified. It also provides a recommendation of alternative packages suitable to the user, and allows prioritisation of requirements in levels of importance. This paper provides an insight into the development of SimSelect, in addition to the reasoning behind the system.

1 INTRODUCTION

Increasing popularity of simulation techniques has resulted in an increase in the number of simulation packages available on the market. The selection of an inappropriate package can result in significant financial losses as well as the disruption of simulation projects. However, appropriate assistance in simulation software selection might reduce the scale of such problems.

Many studies have taken place concerning the selection of simulation software. Pidd (1989) warns of several facts that the potential user should be sceptical about. For example, one should not believe any vendor who claims that his product is better than everyone else's for any application or that the software can run on any computer under any operating system. In addition, when asked about the support they can provide in the case of problems cause by bugs, the majority of vendors would probably deny the possibility of the existence of bugs. Furthermore, the author claims that the type of simulation software to be chosen depends on the intended application, and discusses which basic facilities should be provided in simulation software. Finally, general advice for simulation software selection is provided, which includes: development of a preliminary model of application, consideration of available resources and future applications, examination of the available software, and asking the vendors for assistance if possible.

In a more recent survey Davis and Williams (1995) illustrate the evaluation and selection of simulation software using *the analytic hierarchy process method.* They evaluate five simulation software systems using this method in order to recommend suitable simulation software for a UK company. The chosen criteria include: cost, comprehensiveness of the system, integration with other systems, documentation, training, ease of use, hardware and installation, and confidence related issues (mainly regarding the reputation of the vendor). The conclusions outline that the method used should be considered only as a decision aid, although the outcome of the results was considered to be satisfactory.

Many other studies have been carried out involving the selection of simulation software such as Bovone et al (1989) and Holder (1990). All of these studies are geared towards a more theoretical approach of the problems involving the selection of simulation software. SimSelect attempts to put these issues into practice and aid the user in various difficulties that may arise when selecting a suitable simulation package.

This paper is structured as follows. Subsequent to a description of the criteria used for software evaluation, and justifications for choosing criteria, a description of the database, the way in which the database handles a query and the ease with which it may be modified are provided. Later sections provide an insight into the system itself. Issues related to an overview of the system, requirements specification, and the way in which a query is processed by SimSelect are addressed. The method which SimSelect uses to recommend alternative packages is also discussed. Finally, conclusions outline the main findings of this research, highlight the benefits that can be derived from SimSelect as well as its future developments.

2 CRITERIA

The type of criteria required to evaluate simulation software was an important issue that was addressed throughout the development of SimSelect. Research within this topic was undertaken and several important studies were focused upon (Tocher 1965), (Cellier 1983), (Banks et al. 1991), and (Law and Kelton 1991). However, majority of these studies were theoretical and unsuitable as not many of them provided a critical evaluation of the software products under consideration. In addition, it was difficult to compare the evaluation of different studies because they did not use common criteria.

The criteria used for the SimSelect was taken from Hlupic (1993) who proposed more than 330 evaluation criteria. When considering the SimSelect prototype, it was not feasible to include all these different criteria. It was decided to use about 40 different criteria which ranged from the following groups: General Features, Visual Aspects, Coding Aspects, Efficiency, Modelling Assistance, Testability, Input/Output, Software Compatibility, Experimental Facilities, Statistical Facilities, and Financial and Technical Features. The main reason for incorporating data from such a wide range of categories, is to ensure that SimSelect will be able to produce accurate results.

The criteria used to evaluate a simulation package are not subjective. They are not considered as a rating given on the quality of the features that a simulation package offers. They merely indicate whether certain features are provided or not. It was decided to develop SimSelect to operate in this manner, otherwise the system may have been swayed by the personal prejudices.

3 DATABASE DESIGN

The database is considered to be the "engine-room" of SimSelect. It holds the information related to evaluation details of each package. In addition, the queries concerning the suitability of packages are also generated within the database. The initial requirements of the database were largely concerned with not only the ability to store and access data, but also to ensure that the database could be easily modified and maintained. This is an issue of great importance because in the near future the number of packages which are evaluated will be increased. The data has been collected over several years of research in simulation software evaluation by the first author. Presently, the information included within the database relates to specific features that a simulation package is able to offer. The criteria eliminates any bias as they do not measure how good a certain feature is. They merely record if a feature is available or not.

The database was designed using the Access Engine within Visual Basic 3.0, but was initialised using Acess 1.1. The main reason for implementing the database in this manner, rather than using the Access engine within Visual Basic from the onset, was due to problems that would have occurred with regards to the modification of the database at a later stage. For example, using just the Access engine to initialise the database would have meant that it would not have been possible to delete fields or tables at a later stage.

The database is designed in a conventional manner, with information being stored about each package. A particular data table shown in Figure 2 contains information regarding the simulation software package such as type of package, purpose of simulation, and price. Another table designed, holds information regarding the name of the package and a unique index which holds the pack_id number.

A query is generated after the user has finalised the requirements and priority rating. SimSelect generates a query by splitting the criteria into two sieves. The first sieve contains those features which are termed as General, and are specific to the system. Factors included within this group include type of package, cost of system, type of simulation to be carried out. Regardless of the amount of features that a package may offer, if any of the above features do not meet the requirements of the user, the package can not be considered suitable for the user.

At this stage there can be no compromise in terms of satisfying users General requirements. For example, during the requirements specification, a user specifies that the system should be a general purpose data driven simulator, which can handle continuous simulation, and cost $5,000. SimSelect would only consider those packages which met these general requirements. Should there be no package that meets these criteria a message would appear on the screen as shown in Figure 1. However should a suitable package which matches these requirements be found, the second half of the query is then performed. This involves determining which of the remaining packages matches the requirements specified by the user.

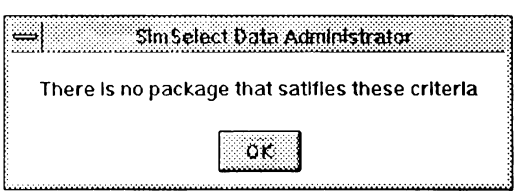

Figure 1: Outcome of an Unsuccessful Query

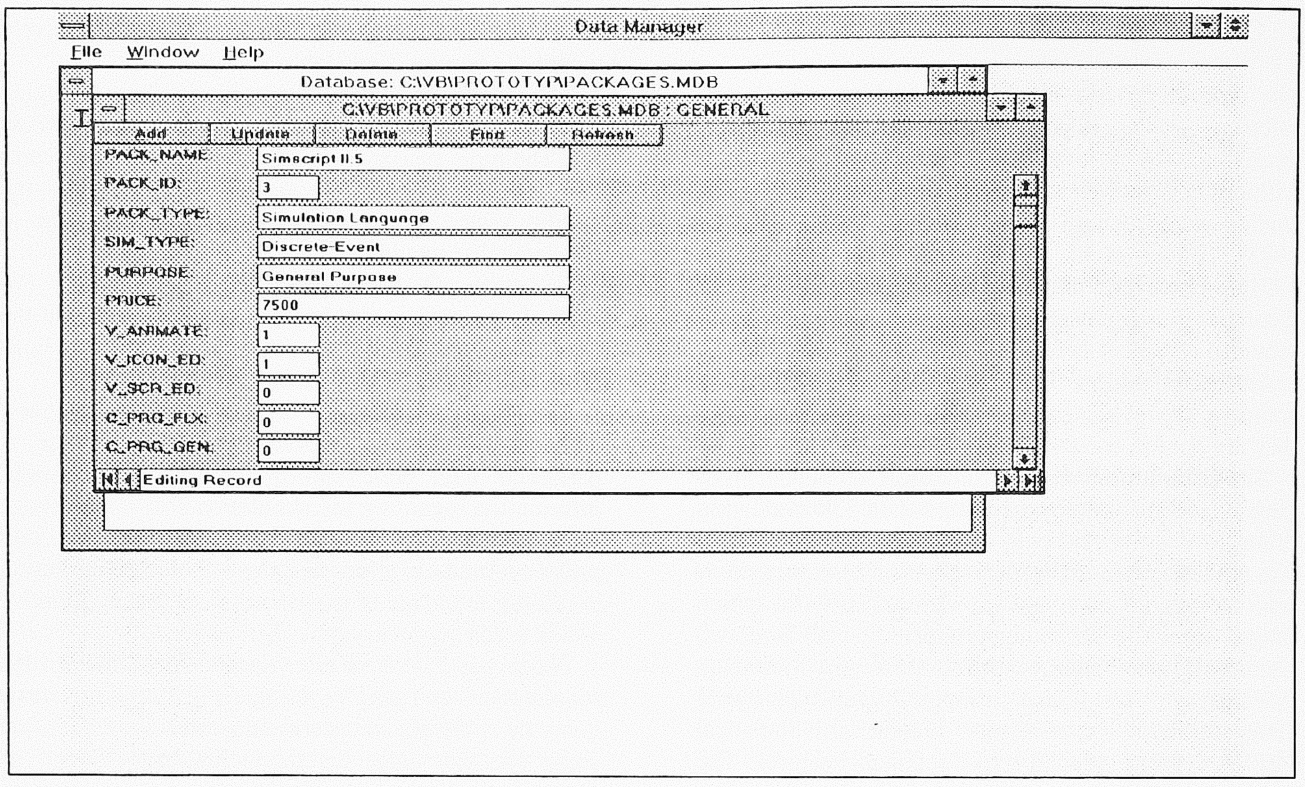

Figure 2: One of the Tables within the SimSelect Database

The priority rating is also an important issue. SimSelect takes it into account when browsing the database in order to find a suitable package. Depending on the number of packages which meet the requirements specified, SimSelect will utilise a pre-built formula stored into the query. In the event that a number of packages suit the criteria, SimSelect will list all of them, otherwise only one package will be recommended. SimSelect also performs another query when the user specifies that the option of Alternative Suggestions should be chosen. For such a choice, the query is formulated by listing all of the packages which were suitable after the first sieve of the query was implemented. The simulation packages are then displayed in an orderly manner. The package which suited a majority of the user's specifications is listed first, with the final package in the list being that which matched just the General Requirements. It is feasible that the database may have to generate possibly three queries depending on the selection choices of the end-user. One concept which was therefore important was to minimise redundancy. Had steps not been taken to insure this, the database may have been easily prone to corruption and be too time consuming when dealing with queries. In order to eradicate this problem, we have built entity relationship diagrams, formalised the data, and constructed the data tables.

4 OVERVIEW OF SIMSELECT

SimSelect was designed using Visual Basic 3.0, because it allowed for the creation of effective and user-friendly interface. Visual Basic 3.0 has a built in Access Engine which is easily capable of handling all database queries. This was ideal for SimSelect which required a relational database to store information regarding various packages.

At the moment, SimSelect stores information about 20 packages, based on 40 different criteria. There is no bias towards a specific supplier, which is one reason why SimSelect may be seen as an effective tool in the simulation software selection process.

5 GENERATING A QUERY

In order to generate a query using SimSelect, the user is required to input data regarding the requirements of the desired system. The main menu of the system is shown in Figure 3. The option provided by the menu are Requirements, Process, Help or Exit.

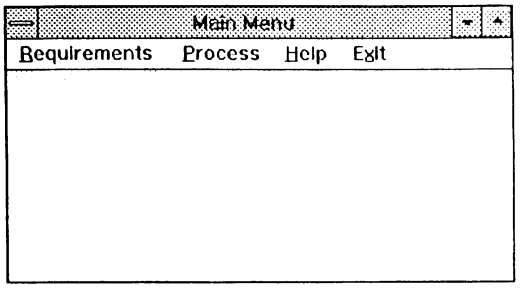

Figure 3: SimSelect Menu System

Once the user has chosen the option marked Requirements, a list comprising the criteria to be entered is displayed in the groups as shown in Figure 4. The user has the choice to enter either all or some of the software requirements.

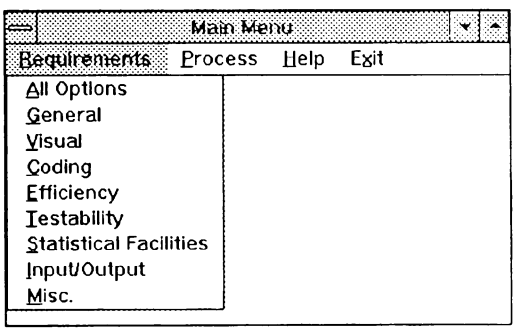

Figure 4: Requirements Option

The main point of reasoning behind this was that it may not be necessary for the user to specify various categories which are not relevant. For example, an end user may wish to focus on only the Visual, Coding and Statistical Facilities of a package and may have no interest in Efficiency or Input/Output facilities. In this case the query is only generated taking the chosen factors into account. However, it is compulsory for the user to select the option regarding General Features for reasons mentioned earlier. This type of information is very influential in deciding whether a package is up to the task of what is required. Figure 5 shows an example of requirements to be selected by the user.

The example shown in Figure 5 relates to the type of package. The user has to decide whether it is necessary for a simulation language to be considered, or a data driven simulator with or without additional programming. Once the information relevant for a particular query has been specified, SimSelect returns to the main menu.

At present, SimSelect is unable to handle any user input other than responses from the options provided. Therefore, should the user make an incorrect choice, SimSelect is unable to detect this. This is a feature which is currently being researched and is to be modified in the near future.

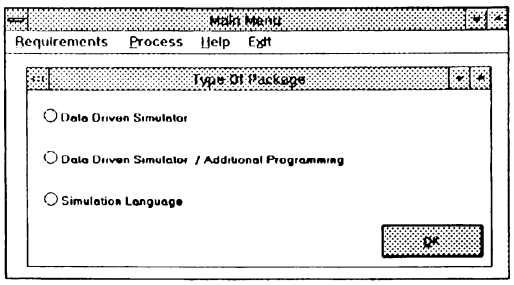

Figure 5: An Example of the Option Concerning General Features of Desired System

6 PROCESSING A QUERY

Following specification of the system requirements a query is generated to determine whether the database contains a simulation package which meets the specifications. To generate a query the user is required to select the process option from the Process menu. This option as shown in Figure 6, enables the user to review all of the options which have been selected. It is possible to change an option by simply clicking on the criteria which requires amendment. On doing so SimSelect allows the user to amend that option in the same way with which it was specified. After completion of the amendment the system returns the user to the Process Option Menu. Once satisfied that the requirements of the desired system are sufficient, it is necessary for the user to prioritise the requirements in terms of levels of importance, as shown in Figure 7.

In order to ensure that a particular feature is included in the desired system, it is possible to give that particular requirement a priority rating. There are three basic ratings which the user can choose from *Low*, *Medium*, and *High*. Should the user choose high priority on a number of ratings, this will ensure that should a suitable package be found, then those criteria with a high priority rating would be included. "Priority Ratings" could be specified for all of the features except the General Features. The calculation required to prioritise the features was designed in such a way as to slightly affect the final judgement regarding which package should be chosen. Thus for a feature assigned a high rating the value was increased from 1 to 1.5, a medium rating amounted to no change and a low rating caused a decrease from 1 to 0.5. Following the assignment of the "Priority Ratings" the user has to progress with the query. Any package included in the requirements scheme but not assigned a priority weighting is automatically defaulted to *Low*.

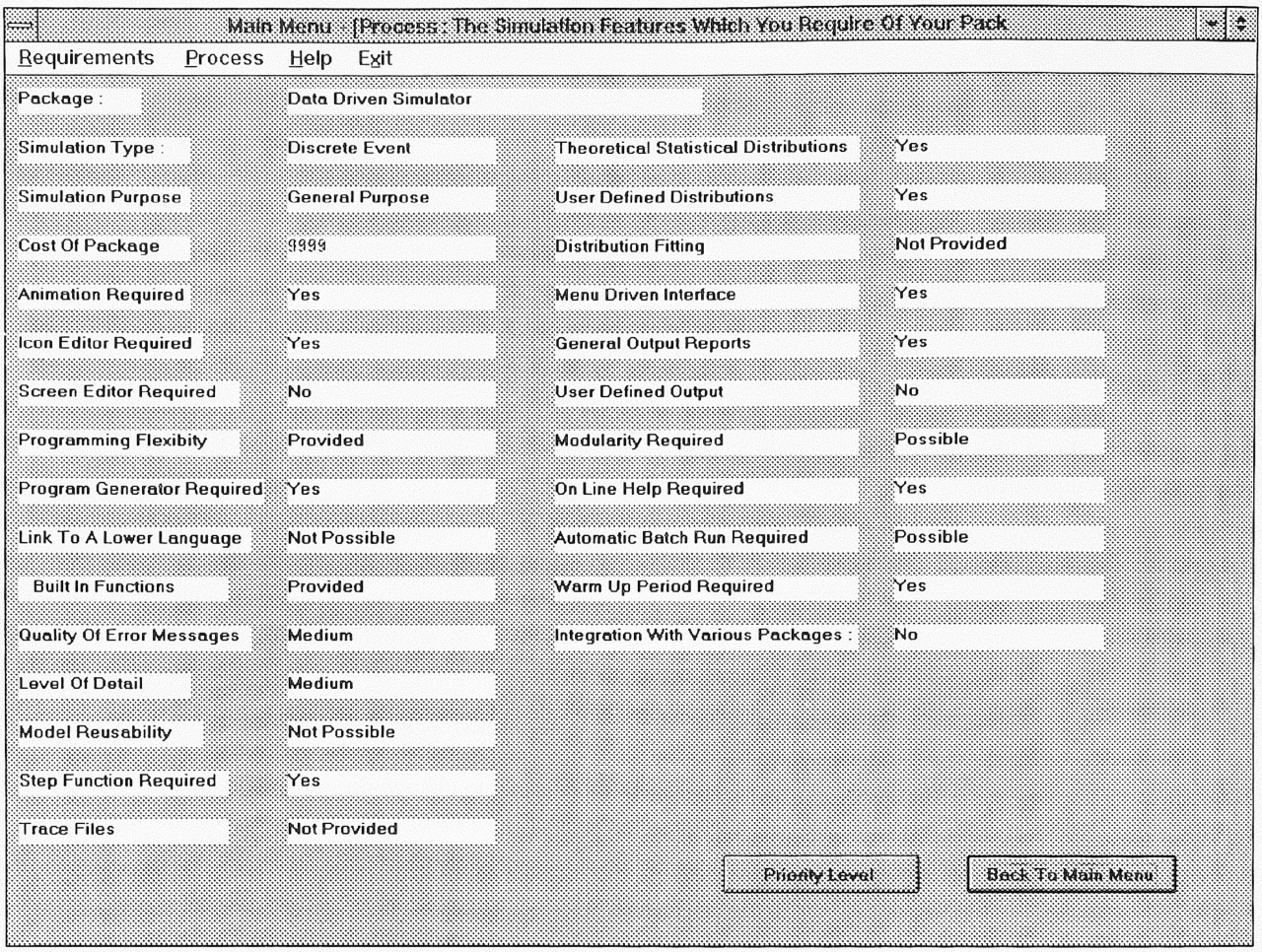

Figure 6: The Process Option Menu

Similarly any requirement not specified at the requirement stage is automatically eliminated at the priority stage. For example, should the user not be concerned with the concept of a "warm-up period" and disregard it at the requirements specification stage, this particular criteria will not be available for selection at the Priority stage. SimSelect is thus able to distinguish such discrepancies as they arise.

Once a query has been generated the user is faced with one of two possible scenarios. The first scenario shown in Figure 2, indicates that a query has been unsuccessful. The second scenario shown in Figure 8 highlights the package which matched the requirements specified by the user. An option available to the user is to view the exact features that the simulation software package contains as shown in Figure 9. By opting for the explanation command at this point, an explanation concerning why this particular package was chosen is displayed

Another operation the user is able to carry out is that of opting for the suggest alternative command. By choosing this option, a box is displayed in the right hand corner of the screen, presenting the user with a list of alternatives which matched the General Requirements of the desired system but may not have exactly matched the other criteria specified. Similarly an explanation at the same stage is given as to why these packages were not recommended in the first instance, but were moreover regarded as alternatives which may be of use to the user.

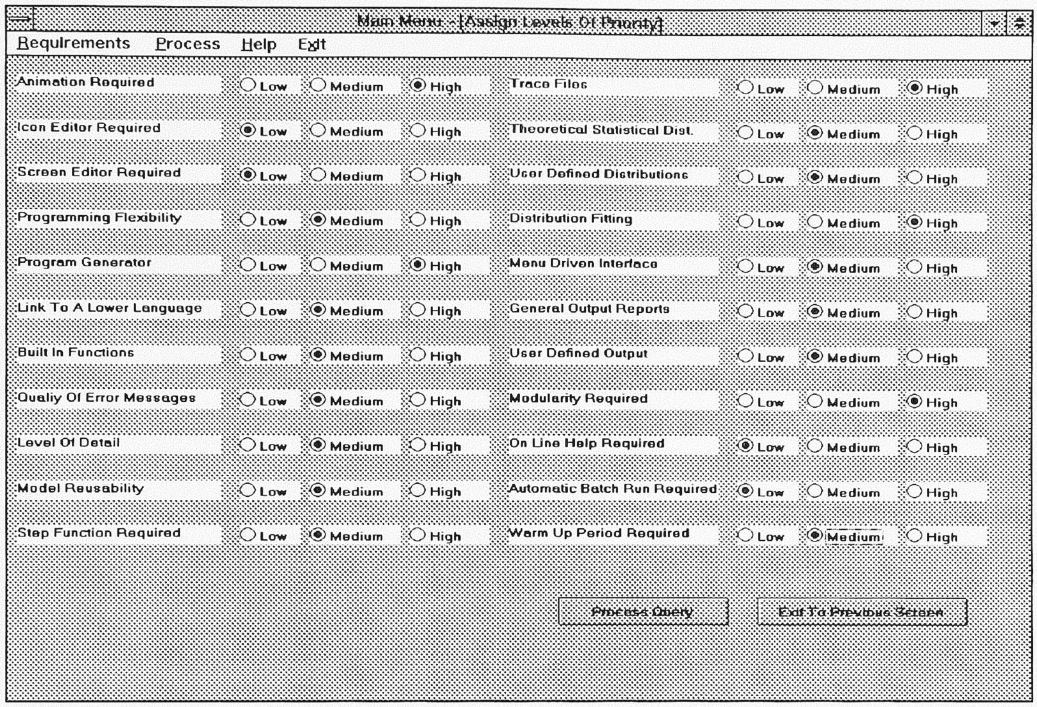

Figure 7: Layout of the Priority Option Screen of SimSelect

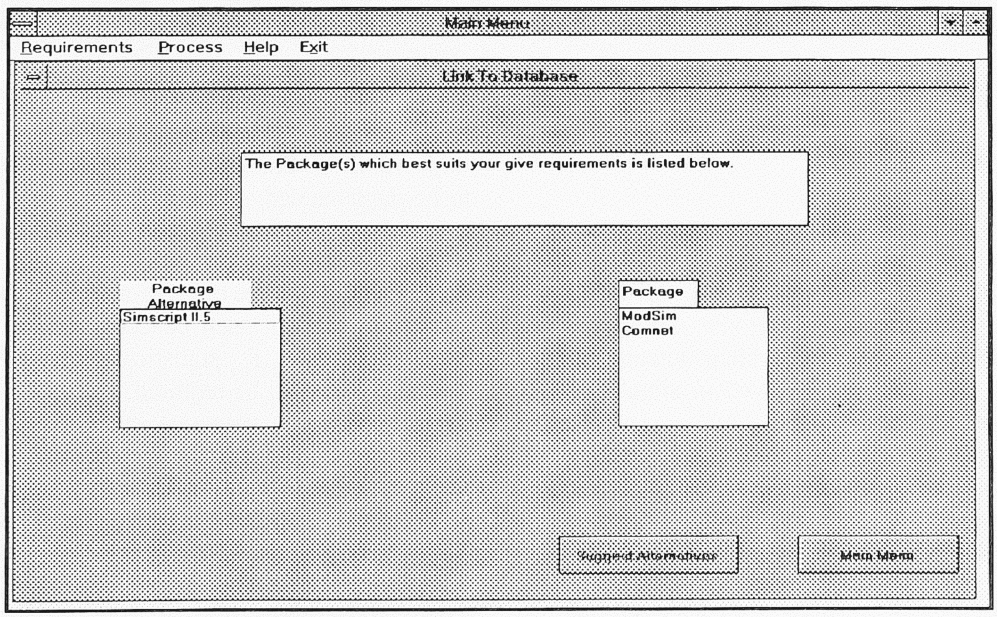

Figure 8: Layout of Recommended Package Screen

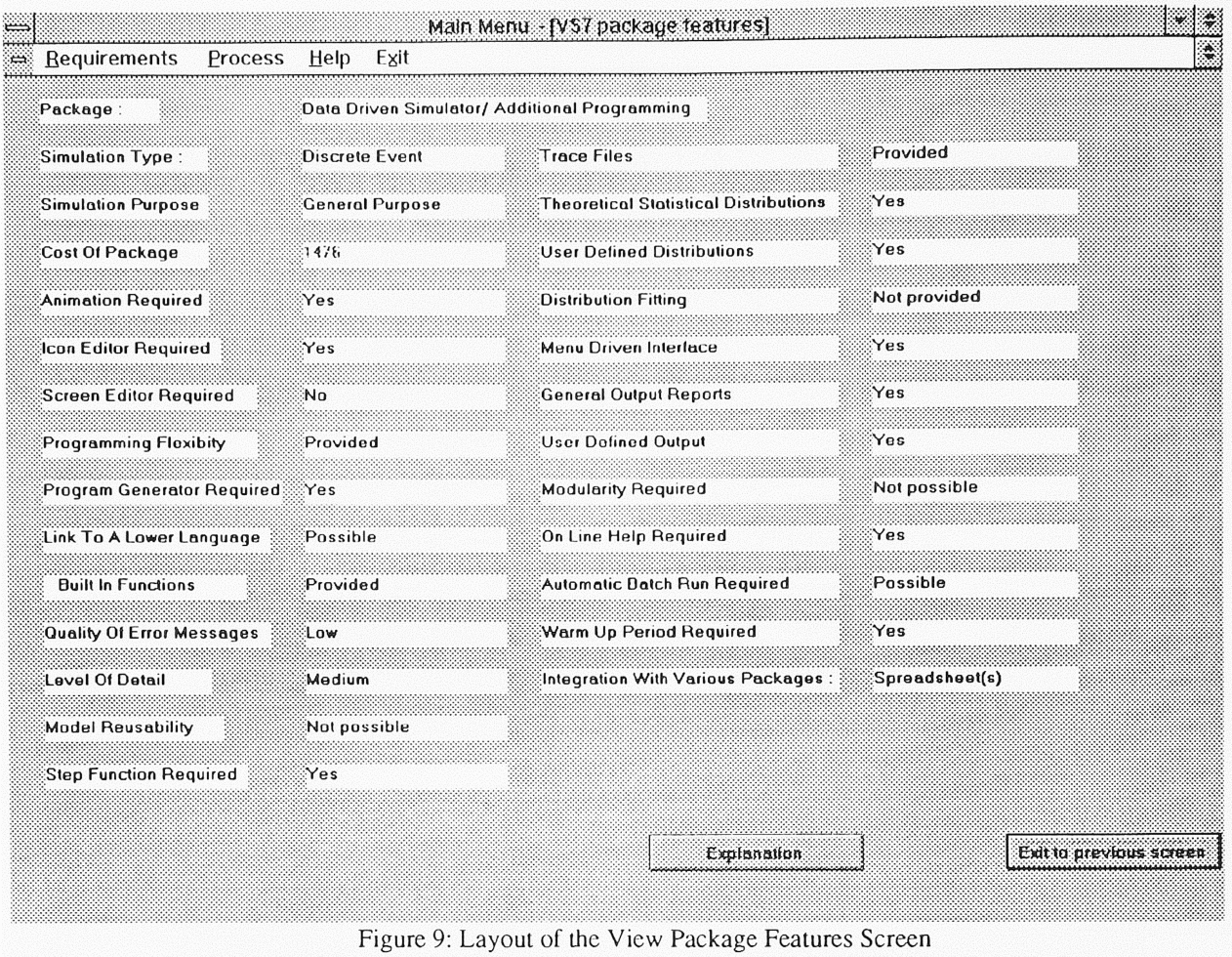

Figure 9: Layout of the View Package Features Screen

7 CONCLUSIONS

SimSelect is a tool which can assist the user in the simulation software selection process. Although the system may seem limited in terms of the number of packages currently evaluated and stored within the database, it still represents an explicit attempt to combat some of the problems involved in simulation software selection.

SimSelect offers an alternative view of selecting simulation software as it is unbiased and unrelated to any software vendor or supplier. Each simulation package has been evaluated and stored in the database against criteria which covers a variety of issues. The criteria does not favour a single package, and the database will be increased to cover more criteria and simulation packages in the near future.

In addition to further expansion of the database, SimSelect will be distributed to various educational and commercial institutions involved in simulation, in order to get more feedback on possible improvements. SimSelect also requires enhancements to be implemented which will allow for the system to actually distinguish between the various levels of criteria that a simulation package can offer. One way to do this would involve creating another data table which takes into account such concepts. For example, after specifying that a statistical distribution fitting mechanism would be required, the user would be able to distinguish how many statistical distributions would be sufficient. The system would then take this into account rather than just offer a package which offered a statistical distribution fitting mechanism but not knowing how many distributions were provided.

Similarly an enhancement is required to allow SimSelect to take into account user comments at various stages so that the user actually inputs information rather than to select from a number of choices offered by the system. The user should be able to input comments regarding the type of animation that is required, the level of coding.

Despite the limitations, SimSelect represents a step forward to more explicit assistance in the simulation software selection process.

REFERENCES

Banks, J., E. Aviles, J.R. McLaughlin, and R.C. Yuan. 1991. The Simulator: New Member of the Simulation Family. *Interfaces* **21**(2):76-86.

Bovone, M., V. De Ferrari, and R. Manuelli. 1989. How to Choose Useful Simulation Software. In *Proceedings of the 1989 European Simulation Multiconference*. 39-43. Simulation Computer Society, San Diego.

Cellier, F.E. 1983. Simulation Software: Today and Tomorrow. In *Proceedings of the IMACS International Symopsium*, ed. J. Burger. and Y. Jarny. 3-19. Elsevier Science Publishers, Amsterdam.

Davis, L., and G. Williams. 1995. Evaluating and Selecting Simulation Software Using the Analytic Hierarchy Approach. Forthcoming in *Integrated Manufacturing Systems*.

Holder, K. 1990. Selecting Simulation Software. *OR Insight*, **3**(4):19-24.

Hlupic, V. 1993. Simulation Modelling Software Approaches to Manufacturing Problems. PhD Thesis, Department of Information Systems, University of London, London, UK.

Law, A.M., and W.D.Kelton. 1991. *Simulation Modelling and Analysis*. 2nd ed. Singapore: McGraw-Hill.

Pidd, M. 1989. Choosing Discrete Simulation Software. *OR Insight*, **2**(3):22-23.

Tocher, K.D. 1965. Review of Simulation Languages. *Operational Research Quarterly*, **16**(2):189-217.

AUTHOR BIOGRAPHIES

VLATKA HLUPIC is a Lecturer in Simulation Modelling in the Department of Computer Science and Information Systems at the Brunel University. She received a Dipl.Econ. and an M.Sc. in Information Systems from the University of Zagreb, and a Ph.D. in Simulation Modelling at the London School of Economics, England. She is researching into and has published extensively, in simulation modelling software approaches to manufacturing problems. She has practical experience in manufacturing and waste disposal industries, as well as having held a variety of teaching posts in England and Croatia. Her current research interests are in manufacturing simulation, business process re-engineering, simulation software evaluation, and in simulators and simulation languages.

AMARDEEP SINGH MANN is a research assistant in the Department of Computer Science and Information Systems at the Brunel University. He received a B.Sc. in Computing Studies from Southbank and an M.Sc. in Simulation Modelling from Brunel University, England. His research interests are in the automisation of the simulation modelling process, software evaluation, and simulation software selection process.