Int J Adv Manuf Technol (1999) 15:366–382
© 1999 Springer-Verlag London Limited

The International Journal of
Advanced
Manufacturing
Technology

Evaluation Framework for Simulation Software

V. Hlupic, Z. Irani and R. J. Paul

Brunel University, Department of Information Systems and Computing, Uxbridge, UK

An increase in the use of simulation as a modelling and analysis tool has resulted in a growing number of simulation software products on the market. Because of this, the importance of an adequate approach to simulation software evaluation and selection is apparent. This paper presents a comprehensive framework which can be used for evaluation of simulation software packages. The evaluation criteria are grouped according to their nature, and they can be of practical use to anyone involved in simulation software evaluation and selection.

Keywords: Evaluation criteria; Simulation; Simulation software

1. Introduction

One of the major uses of simulation is in the planning of new manufacturing operations. Wild and Otis [1] describe the case of a company that considered the implementation of 77 new machine tools, for a new production line. However, when the operation was simulated, they found that 4 machines were actually needed, presenting a total saving of $750 000. Such savings are not unusual when simulation is used.

Simulation modelling is being applied to many areas such as manufacturing, material handling, health care, military, transportation, business process re-engineering and computer system performance [2]. As a result, simulation is often used to make current operations more efficient, resulting in substantial savings. It is often used to help production systems become more efficient, with better on-line deliveries and less investment. Simulation allows many variables to be considered simultaneously. It can be used to solve problems that are too complex to analyse otherwise. Therefore, modelling the system, through its simulation, and then experimenting with it, leads to a better understanding of its operation [3]. Simulation modelling can also be used as a tool for employee training and troubleshooting. As described by Wilkinson [4], Polaroid's line supervisors are taught how to use simulation for identifying possible, or probable problems, thus allowing the development of appropriate corrective action. However, management are often hesitant to accept the results of simulation. They are more comfortable with more familiar methods and may not believe that the simulation accurately represents their production system. Simulation in the past presented results in the form of large amounts of data, however, with modern use, simulation can be presented to senior management in understandable terms, for example, animated graphics.

Advances in technology, have allowed simulation to become a more widely available tool. In part, this is facilitated by the price of simulation software, which is more affordable to a greater number of companies. Also, simulation languages are more user-friendly. However, users need advice on how to select an adequate simulation software tool and which software features to examine during the evaluation process.

This paper presents a large number of evaluation criteria derived for the evaluation of simulation packages, grouped according to their nature. Some of the criteria necessarily overlap, for example, ease of use and documentation. It may be debatable, therefore, why a specific criterion is included in one subsection and not in another. There are also some criteria that are more general, comprising several specific criteria. For example, ease of use of software depends on many factors, such as the quality of documentation, on-line help, tutorials and the experience of the user. However, most of these criteria are listed separately to emphasise their importance. In addition, the aim was to derive a comprehensive evaluation framework that can be of practical use, rather than to invent a strict methodological classification of criteria.

In addition to an evaluation framework, a review of research studies in simulation software evaluation criteria is provided. The main findings of these studies are analysed and their context for this research presented. The possible use of the derived criteria is briefly discussed and the methodology for simulation software selection is presented. The main findings of this research are summarised and conclusions subsequently drawn.

Correspondence and offprint requests to: Dr Z. Irani, Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK. E-mail: zahir.irani@brunel.ac.uk

2. Background Research Material

Establishing simulation software evaluation criteria has been the subject of much research. For example, Kochhar and Ma [5] address the essential and desirable features of simulation software for its effective use in manufacturing environments, providing criteria which should be used for the selection of manufacturing simulation software tools. These criteria relate to: modelling assistance; interactivity; graphics; a data handling capability; the time scale for model development; the learning curve and the required skills for the use of software; ease of model editing; portability; simulation speed; and interfacing the simulation package with external systems.

In addressing the issues relating to the choice of simulation software products for the analysis of manufacturing systems, Haider and Banks [6] establish the following desirable features for simulation software: input flexibility; structural modularity; modelling flexibility; macro capability and hierarchical modelling; materials handling modules; standard statistics generation; data analysis; animation; interactive model debugging; micro/mainframe compatibility; the support provided by the supplier; and the cost of simulation software.

In discussion of the role of simulation in designing and scheduling manufacturing systems, Grant [7] provides a list of features which manufacturing simulators should possess. Some of these features include an effective user interface; an implemented set of algorithms for sequencing production orders; interactivity; an interface to external data sources; a mechanism to store all input and output data in a database; and fast execution of simulation; coded algorithms; standard and user-coded performance reports; storage of data used for model design in an external file; and orientation to the design issues including randomness. Kochhar [8] presents criteria for the assessment of manufacturing simulators. These criteria include: the world views adopted by the simulator; modelling assistance provided; interactive capability; animation facilities; a data handling capability; learning curve; ease of use; portability of simulation software; simulation speed; reliability and service; flexibility and facilities for data recording and output results.

Bright and Johnson [9] discuss the intrinsic nature of visual interactive modelling (VIM) software. Three main features of this type of software have been addressed: speed and adaptability; width of application; and ease of use. Szymankiewicz et al. [10] specify desirable features of simulation software. The main features which should be more extensively used in simulation software include: suitability for a wide range of problems; portability; lower price; ease of learning; additional programming; user-friendliness; integration with real-life control systems; built-in debugging facilities; high-resolution graphics; panning and zooming; statistical facilities for multiple runs; and a breakpoints setting capability.

Mackulak and Savory [11] carried out a questionnaire survey on the most important simulation software features. The most important features identified include: a consistent and user friendly user interface; database storage capabilities for input data; an interactive debugger for error checking; interaction via mouse; a troubleshooting section in the documentation; storage capabilities for simulation models and results; a library of reusable modules of simulation code; and a graphical display of input and output. Banks et al. [12] proposed the following groups of features to consider when selecting a simulator: input features; processing features; and output features.

The above studies show which features of simulation software are considered as important and should be incorporated into simulation software, and/or could be used for software evaluation. It is evident that in software evaluation, many factors should be assessed, and their significance weighted in order to evaluate, compare and select adequate software.

None of the studies analysed provides an extensive list of criteria for the evaluation of simulation packages. This paper presents over 250 evaluation criteria and provides considerably more comprehensive guidelines for the assessment of simulation software packages than those found in the literature.

3. Simulation Software Evaluation Criteria

The criteria derived can be applied to the evaluation of any general or special purpose simulation package. They are placed into 13 groups of criteria according to their nature. Owing to the comprehensiveness of the evaluation framework, individual criteria within each group are merely listed, and generally described in the context of a particular group. A classification of simulation packages in the context of the derived criteria is also provided. According to the type of each criterion, the classification determines whether, for example, a certain feature exists in the package, determines the quality of features provided, or lists types of alternatives available within a particular feature. A fuller description of each criterion is provided in Hlupic and Paul [13].

3.1 General Features

Criteria included in this group describe general features of the package. Most of these criteria relate to modelling aspects such as the type of formal logic needed for modelling (if any), the method of changing the state of the model (process based, activity based, event based, three phase, or a combination of these methods), type of simulation (discrete event, continuous or combined), the level of modelling transparency, etc. There are also some criteria that evaluate the level of experience and formal education in simulation required by the user, and examine how easy it is to learn and use the package. Table 1 shows the criteria classified in this group.

3.2 Visual Aspects

Graphical presentations of simulation models and animation of simulation are very important characteristics of simulation software. Criteria included in this group relate to the type and quality of graphical facilities provided by the package. These criteria evaluate, for example, whether it is possible to perform an animation of the simulation experiments, the types of animation provided by the package, and whether it is possible to manipulate icons. Table 2 shows the criteria classified in this group.

Table 1. Criteria for general features.

General features	
Criteria	Classification
1. Type of package	Data driven simulator Data driven simulator with additional programming capability Simulation language
2. Type of simulation	Discrete event Continuous Both
3. Purpose	General purpose Manufacturing oriented Other special purpose
4. Terminology	General terminology Application specific terminology
5. Modelling approach	Process based Activity based Event based Three phase Combination
6. Formal logic	Required Not required
7. Representativeness of models	High Medium Low
8. Ease of conceptualisation of simulation logic	Easy Not easy
9. Modelling transparency	High Medium Low
10. Hierarchical model building	Possible Not possible
11. Run-time applications	Provided Not provided
12. Conceptual model generator	Provided Not provided
13. Versions of software for different operating systems	UNIX Windows 3.1 Windows 95 Windows NT OS/2
14. The length of entity name	Long (e.g. unlimited) Medium Short (e.g. up to 3 characters)
15. Entity name	User defined System defined
16. Experience required for software use	None Some Substantial
17. Formal education in simulation required for software use	None Some Substantial
18. User friendliness	High Medium Low
19. Ease of learning	Easy Not easy
20. Ease of using	Easy Not easy
21. Initialisation	Possible Not possible
22. Specification of time units	Possible Not possible
23. Integration of operations (data gathering, simulation model development, output data analysis)	Provided Not provided
24. Real-time simulation models	Possible Not possible
25. Distributed simulation on network environment	Possible Not possible
26. Specification of length measures	Possible Not possible

Table 2. Criteria for visual aspects.

Visual aspects	
Criteria	Classification
1. Animation	Possible Not possible
2. Type of animation	Full animation Semi-animation (state-to-state)
3. Timing of animation	Concurrent animation Post-processed animation
4. Type of graphical display	Icons Symbols Characters
5. 3D graphics	Provided Not provided
6. Integrity of graphics	Integrated to the package Separate
7. Animation layout development	Concurrent with model development Before model development After model development Flexible
8. Multiple screen layout	Possible Not possible
9. Facility for customising the view of the model	Provided Not provided
10. Playback mode	Provided Not provided
11. Importing graphics and multimedia elements	CAD packages Media Control Interface Bitmap
12. Animation with visual clock	Provided Not provided
13. Icon editor	Provided Not provided
14. Screen editor	Provided Not provided
15. Ease of icon development	Easy Not easy
16. Ease of using screen editor	Easy Not easy
17. Types of icons	Bit mapped Pixel based
18. Icon library	Provided Not provided
19. Merging icon files	Possible Not possible
20. Resizing of icons	Possible Not possible
21. Rotating of icons	Possible Not possible
22. Changing the colour of the icons	Possible Not possible
23. Zoom function	Provided Not provided
24. Panning	Provided Not provided
25. Switching on/off the graphic	Possible Not possible
26. Switching between screens	Possible Not possible
27. Switching between character and icon graphics	Possible Not possible
28. Print screen facility	Provided Not provided
29. Virtual screen	Provided Not provided
30. Indication of the element status	Provided Not provided
31. Changing the colour of the element status display	Possible Not possible
32. Limitation on number of displayed icons	Exists Does not exist
33. Number of icons stored in icon library	Large Medium Small
34. Change of icons during simulation	Possible Not possible
35. Icons with multiple colours	Provided Not provided
36. Virtual reality features	Provided Not provided
37. Easy copying of icons	Possible Not possible

3.3 Coding Aspects

The possibility of additional coding might be a very important feature of a package. This feature determines the flexibility and robustness of the software, which is especially valuable when complex systems are to be modelled. Criteria included in this group determine whether the package allows additional programming, if access to the code is possible, the characteristics of the added code, the programming concepts supported, etc. Table 3 shows the criteria classified in coding aspects.

3.4 Efficiency

Criteria classified in this group determine the effectiveness and the power of simulation software. Efficiency is expressed both by the capability of the software to model a variety of complex systems and by the characteristics which can save time needed for modelling, and improve the quality of modelling, such as model reusability, reliability, compilation and execution time and multitasking. Table 4 shows the criteria included in this group.

3.5 Modelling Assistance

Criteria systematised in this group evaluate the type and level of assistance provided by the package during modelling. For example, these criteria examine the comprehensiveness of prompting, on-line help if it is provided, whether the package enables modular model development and writing the documentation notes (this feature enables the writing of documentation concurrently with the model development), and whether the model and data can be separated. Table 5 shows the criteria included in modelling assistance.

3.6 Testability

This group comprises criteria that examine which facilities for model verification are provided by the package. These facilities include error messages, displays of the values of logical elements such as functions and variables, the possibility of obtaining special files for verification such as list, trace and echo files, provision of step function, etc. Table 6 shows the criteria included in this group.

3.7 Software Compatibility

These criteria evaluate whether the package can be interfaced to other software systems, in order to exchange data with these systems. This feature can considerably enhance the capabilities of the package, especially when complex real systems are modelled. Integration with programming languages is not included in this group of criteria, because it is contained in coding aspects. Table 7 shows the criteria included in this group.

3.8 Input/Output

Criteria included in this group investigate how the user can present the data to the package and the type and quality of output reports provided by the package. These criteria evaluate, for example, whether the package has a menu-driven interface, whether static and dynamic output reports are provided, and how understandable these reports are. Table 8 shows the criteria categorised in this group and a classification of the packages regarding input and output of data.

3.9 Experimentation Facilities

Criteria classified in this group evaluate the variety and characteristics of experimentation facilities. These facilities are required for improving the quality of simulation results and for speeding up the process of designing experiments and of the experimentation itself. Table 9 shows the criteria included in this group.

3.10 Statistical Facilities

Owing to the randomness that is present in the majority of simulation models, good statistical facilities are very important. Criteria included in this group examine the range and quality of statistical facilities provided by the simulation package. Table 10 shows the criteria included within user support.

3.11 User Support

The following criteria evaluate the type and quality of user support provided by the software supplier, which can facilitate learning and using the package. These criteria not only include technical support in the form of documentation, and demo disks, but also include a variety of services provided by the software supplier which ease the use of the package and keep the user informed about plans for future software improvements. Table 11 shows the criteria included in this group.

3.12 Financial and Technical Features

Criteria included in this group examine features of the package related to its costs and technical characteristics. Some of the issues considered here are: how expensive is it to purchase a certain package, to install and maintain it, whether any additional hardware would have to be purchased for installation of the package, etc. Table 12 shows the criteria included in this group.

3.13 Pedigree

Criteria included in this group refer to the origin of the package and its importance. They also evaluate how widely the package is used, and judge the reputation of the software supplier. A supplier's reputation is a criterion which depends on many factors such as the length of time the supplier has been

Table 3. Criteria for coding aspects.

Coding aspects		
Criteria		Classification
1.	Programming flexibility	Provided No additional programming provided
2.	Program generator	Provided Not provided
3.	Access to source code	Possible Not possible
4.	Readability of source code	High Medium Low
5.	Readability of added code	High Medium Low
6.	Self-documentation of added code	High Medium Low
7.	Precision of added code	High Medium Low
8.	Comprehensiveness of added code	High Medium Low
9.	Link to a lower language	Possible Not possible
10.	Data storage, retrieval and manipulation facilities	Provided Not provided
11.	Quality of data storage, retrieval and manipulation facilities	High Medium Low
12.	Built-in functions	Provided Not provided
13.	User functions	Possible Not possible
14.	Global variables	Provided Not provided
15.	Names of functions, variables and attributes	User defined System defined
16.	Writing comments for logical elements	Possible Not possible
17.	Type of time variable	Real Integer
18.	Type of translation	Compilation Interpretation
19.	Text/code manipulation	Possible Not possible
20.	Length of the lines in coding editor	Large Medium Small
21.	Support of programming concepts	Provided Not provided
22.	Quality of the support for programming concepts	High Medium Low
23.	Interface to user written programs	
24.	Object oriented programming concepts	Provided Not provided

Table 4. Criteria for efficiency.

Efficiency		
Criteria		Classification
1.	Robustness	High Medium Low
2.	Level of detail	High Medium Low
3.	Number of elements in the model	Large Medium Small
4.	Model reusability	Possible Not possible
5.	Model status saving	Possible Not possible
6.	Automatic saving	Possible Not possible
7.	Interaction	Possible Not possible
8.	Adaptability to model changes	High Medium Low
9.	Multitasking (i.e. Performing several tasks at the same time)	Possible Not possible
10.	Model chaining (i.e. linking outputs from different models)	Possible Not possible
11.	Exit to the operating system within the package	Possible Not possible
12.	Compilation time	Long Medium Short
13.	Model execution time	Long Medium Short
14.	Case sensitivity	Provided Not provided
15.	Conversion of numbers (real vs. integer)	Provided Not provided
16.	Various queuing policies	Provided Not provided
17.	Number of queuing policies	Large Medium Small
18.	Time scale for model building	Large Medium Small
19.	Reliability	High Medium Small
20.	Pre-existing generic models	Provided Not provided
21.	Merging of models	Provided Not provided
22.	Editing partially developed models	Possible Not possible
23.	Automatic model building	Provided Not provided
24.	Ease of model editing	Easy Not easy
25.	Interactive handling of parameters during experimentation	Possible Not possible
26.	Specification of part flow by a mouse	Provided Not provided

Table 5. Criteria for modelling assistance.

Modelling assistance		
Criteria		Classification
1.	Prompting	Provided Not provided
2.	Quality of prompting	High Medium Small
3.	Modularity	Possible Not possible
4.	Model and data separation	Possible Not possible
5.	Facility for designing reusable user defined elements	Provided Not provided
6.	Libraries and templates of simulation objects	Provided Not provided
7.	Warning messages which for operations which affect the model file (e.g. overwriting, closing file not saved)	Provided Not provided
8.	Warning messages for operations which affect model currently developed	Provided Not provided
9.	Context sensitive prompt to facilitate model development	Provided Not provided
10.	Undo/redo commands	Provided Not provided
11.	Automatic connection between elements	Provided Not provided
12.	Use of mouse	Possible Not possible
13.	One-line help	Provided Not provided
14.	Quality of on-line help	High Medium Low
15.	Search facilities within help	Provided Not provided
16.	Help on system messages	Provided Not provided
17.	Printing help text	Possible Not possible
18.	Documentation notes for inserting comments concurrently with model development	Provided Not provided
19.	Quality of facility for documentation notes	High Medium Low
20.	Text editor as integral part of the package	Provided Not provided
21.	Automatic editing of data	Provided Not provided

Table 6. Criteria for testability.

Testability		
Criteria		Classification
1.	Logic checks	Provided Not provided
2.	Interactive error messages	Provided Not provided
3.	Quality of error messages	High Medium Small
4.	Moment of error diagnosis	Model entry Compilation Model execution Combination
5.	Ease of debugging	Easy Not easy
6.	Display of function values	Possible Not possible
7.	Display of attributes	Possible Not possible
8.	Access to attributes	Possible Not possible
9.	Display of variables	Possible Not possible
10.	Display of element's state	Possible Not possible
11.	Dynamic display of capacity	Possible Not possible
12.	Display of the workflow path	Provided Not provided
13.	Display of events on the screen	Provided Not provided
14.	Display of part position within element	Provided Not provided
15.	Facility for immediate user actions	Provided Not provided
16.	List files (list of model entities and parameters)	Provided Not provided
17.	Echo	Provided Not provided
18.	Trace files (showing events and entity status)	Provided Not provided
19.	Explode function (showing a state of an element)	Provided Not provided
20.	List of used elements	Provided Not provided
21.	Backward clock	Provided Not provided
22.	Step function (event to event jumping)	Provided Not provided
23.	Flow analysis	Provided Not provided
24.	Interactive debugger	Provided Not provided
25.	Display of parts flow tracking record collected during simulation run	Provided Not provided
26.	Audible alarms	Provided Not provided
27.	Rejection of illegal inputs	Provided Not provided

Table 7. Criteria for software compatibility.

Software compatibility		
Criteria		Classification
1.	Integration with spreadsheet packages	Possible Not possible
2.	Integration with statistical packages	Possible Not possible
3.	Integration with word processors	Possible Not possible
4.	Integration with computer-aided design software	Possible Not possible
5.	Integration with database management system	Possible Not possible
6.	Integration with expert systems	Possible Not possible
7.	Integration with manufacturing requirements planning software	Possible Not possible
8.	Integration with scheduling software	Possible Not possible

present in the software market, the number of employees and representative offices the supplier has, and the type and level of user support that is provided. Table 13 shows the criteria included in this group.

4. Use of Criteria

The possible use of the above criteria is investigated from the perspective of software use. Two main types of software use are established: the use of a package for education, and for modelling complex real-life systems. A more detailed discussion on the use of criteria is provided in Hlupic and Paul [14].

4.1 Selection of a Package for Education

It was assumed that the users in this group would use the simulation package mainly for educational purposes, and that the main users would be students, without or with little previous experience in simulation modelling. The use of software at educational institutions for modelling complex real-life problems was not presumed here.

The proposed hierarchy of criteria for the selection of packages for education favours criteria regarding ease of use and learning the package, modelling assistance provided by the package, efficiency, incorporated physical elements typical of manufacturing systems, and visual aspects. In particular, some of the most relevant individual criteria (within important groups of criteria) include: on-line help, prompting, logic checks (modelling assistance); user friendliness, ease of learning, ease of using, modelling transparency (general features); animation, icon editor (visual aspects); interaction, reliability, time scale for model building (efficiency); quality of error messages, step function, list files, display of events on the screen (testability); menu driven interfaces, dialogue boxes, static graphical output (input/output); quality of documentation, tutorials, demo models (user support); educational discount, no security device, version of software for network (financial and technical features), etc.

All criteria that support learning and relatively quick and easy model development have higher importance than those that enable the handling of a large quantity of data (e.g. software compatibility), and detailed modelling (e.g. coding aspects). The main reason for this is the relatively short duration of simulation courses in many cases. Therefore, it is believed that students should not spend too much time on model construction. They should also be able to learn the basics of simulation methodology, such as the use of statistics in simulation, conceptual model development, model validation and verification techniques, design of experiments and analysis of simulation output.

4.2 Selection of a Package for Modelling Real-Life Problems

Selection of a package for modelling real-life problems is divided into two groups according to the modelling purpose. The first group presents a hierarchy of criteria that might be applied for the selection of a package to be used for rapid modelling, whilst the second group establishes a hierarchy of criteria for selection of a package for the detailed modelling of complex real-life problems.

4.2.1 Rapid Modelling

For this type of modelling it is assumed that users have some previous experience in simulation modelling and they know its basic methodological issues. Rapid modelling of real systems means that models should be developed as quickly as possible without too many details in order to provide basic information about the system being modelled. The hierarchy of criteria for the selection of packages for rapid modelling shows that the most important criteria are, modelling assistance for easy development of models, and efficiency for saving the time needed for model development. These criteria are more important than those that facilitate detailed and complicated modelling with a large quantity of data such as criteria concerning coding aspects, software compatibility, and extensive statistical and experimental facilities.

Examples of the most important individual criteria include: ease of using, modelling transparency (general features); animation, icon library (visual aspects); automatic editing of data, prompting, logic checks, use of mouse (modelling assistance); interaction, time scale for model building, automatic model building, model reusability, compilation time, execution time (efficiency); menu driven interface, user defined output, dynamic graphical output (input/output); number of scheduling rules provided, conditional routeing (scheduling features); quality of error messages, display of events on the screen (testability), etc.

Table 8. Criteria for input/output.

Input/output		
Criteria		Classification
1.	Menu driven interface	Provided Not provided
2.	Pull down menus	Provided Not provided
3.	Type of menu selection	By mouse By keys Other
4.	Selection buttons	Provided Not provided
5.	Dialogue boxes	Provided Not provided
6.	Multiple inputs	Possible Not possible
7.	Model input	Interactive Batch mode
8.	Database maintenance for input/output	Provided Not provided
9.	Multiple outputs	Possible Not possible
10.	General output reports	Provided Not provided
11.	Static graphical output	Provided Not provided
12.	Dynamic graphical output	Provided Not provided
13.	Types of graphical display	Bar graphs Histograms Level graphs Pie charts Line graphs Scatter diagrams Time series Area graphs
14.	User defined output	Possible Not possible
15.	Automatic rescaling of histograms and time series	Provided Not provided
16.	Quality of output reports	High Medium Low
17.	Understandability of output reports	High Medium Low
18.	Periodic output of simulation results	Provided Not provided
19.	Availability of results before end of simulation	Provided Not provided
20.	Input data reading from files	Provided Not provided
21.	Writing reports to files	Provided Not provided
22.	Writing reports to printer	Provided Not provided
23.	Writing reports to plotter	Provided Not provided
24.	Snapshot reports	Provided Not provided
25.	Summary reports for multiple runs	Provided Not provided

Table 9. Criteria for experimentation facilities.

Experimentation facilities		
Criteria		Classification
1.	Automatic batch run	Possible Not possible
2.	Warm-up period	Provided Not provided
3.	Independent replications of experiments	Provided Not provided
4.	Re-initialisation	Provided Not provided
5.	Re-start from non-empty state	Possible Not possible
6.	Breakpoints	Provided Not provided
7.	Speed adjustment	Provided Not provided
8.	Experimental design capability	Provided Not provided
9.	Quality of experimental design facility	High Medium Low
10.	Accuracy check	Provided Not provided
11.	Automatic determination of run length	Provided Not provided

4.2.2 Detailed Modelling

When a simulation study is carried out to develop detailed models of complex real-life systems, then the most important criteria are those symbolising the power of the package regarding its robustness, modelling flexibility and efficiency. It is assumed that users of software in this group have experience in simulation modelling and a certain level of theoretical knowledge about simulation. The hierarchy of criteria for the selection of a package for detailed real-life problems has the largest number of criteria that have the highest level of importance in comparison to the other two hierarchies.

The most relevant criteria are those regarding the flexibility of the package supported by coding aspects, the possibility of integration with database management systems to handle a large quantity of data, efficiency that can speed up detailed and complex modelling, and testability which can ease the time-consuming process of model verification. For this type of modelling, it is also important for the package to provide good support in experimentation and in statistical facilities, and the possibility of obtaining special user-defined reports. Issues related to scheduling are also important, as well the ability to quickly model a variety of physical elements and operations in systems under consideration.

Some of the most important individual criteria that should be used for the evaluation of simulation software used for detailed modelling are: programming flexibility, link to a lower language, data storage, retrieval and manipulation facilities (coding aspects); integration with DBMS, integration with CAD software, integration with expert systems, integration with statistical packages (software compatibility); robustness, level of detail, number of elements in the model, reliability, model reusability, model chaining, merging of models (efficiency); animation, virtual screen, multiple screen layout, icon editor, no limitation on number of displayed icons (visual aspects); quality of error messages, trace files, display of variables, display of functions, list files (testability); special user-defined reports, input data reading from files, writing reports to files, summary reports for multiple runs (input/output); automatic batch run, experimental design capability, accuracy check (experimentation facilities); logic check, model and data separation, automatic editing of data, documentation notes (modelling assistance), etc.

Table 10. Criteria for statistical facilities.

Statistical facilities		
Criteria		Classification
1.	Theoretical statistical distributions	Provided Not provided
2.	Number of theoretical statistical distributions	Large Medium Small
3.	User-defined distributions	Possible Not possible
4.	Random number streams	Provided Not provided
5.	Number of different random number streams	Large Medium Small
6.	User-specified seeds of random number streams	Provided Not provided
7.	Antithetic sampling	Provided Not provided
8.	Distribution fitting	Provided Not provided
9.	Goodness-of-fit tests	Provided Not provided
10.	Output data analysis	Provided Not provided
11.	Quality of data analysis facility	High Medium Low
12.	Confidence intervals	Provided Not provided

4.3 Methodology for Selecting Simulation Software

The above presented evaluation framework can be used as a part of simulation software selection methodology [13]. This methodology consists of the following six stages to be followed in the process of simulation software selection.

Table 11. Criteria for user support.

User support		
Criteria		Classification
1.	Documentation (manuals)	Provided Not provided
2.	Quality of documentation	High Medium Low
3.	Reference card	Provided Not provided
4.	Glossary	Provided Not provided
5.	Technical and promotional information material (e-mail, bulletin board)	Provided Not provided
6.	Discussion groups on the Internet	Provided Not provided
7.	Lecturer's guide for educational licences	Provided Not provided
8.	Demo disks	Provided Not provided
9.	Tutorial	Provided Not provided
10.	Training course (basic, advanced)	Provided Not provided
11.	Custom tailored training course	Provided Not provided
12.	Duration of training courses	Long Medium Short
13.	Frequency of training courses	Frequent Not frequent
14.	Demo models	Provided Not provided
15.	Help-line	Provided Not provided
16.	User group meetings	Provided Not provided
17.	Frequency of user group meetings	Frequent Not frequent
18.	Newsletter	Provided Not provided
19.	Package maintenance	Provided Not provided
20.	Consultancy	Provided Not provided

Table 12. Criteria for financial and technical features.

Financial and technical features	
Criteria	**Classification**
1. Portability	Provided Not provided
2. File conversion	Possible Not possible
3. Price	High Medium Low
4. Installation costs	High Medium Low
5. Ease of installation	Easy Not easy
6. Hardware requirements	High Medium Low
7. Availability of package on standard hardware	Provided Not provided
8. Availability of package on standard operating systems	Provided Not provided
9. Version of software for network	Provided Not provided
10. Virtual memory facility	Provided Not provided
11. Security device	Needed Not needed
12. Free software trials	Provided Not provided
13. Free technical support	Provided Not provided
14. Types of contracts available	Many Not many
15. Educational discount	Provided Not provided
16. Quantity discount	Provided Not provided
17. Life cycle maintenance costs	High Medium Low
18. Price of training course	High Medium Low
19. Consultancy fees	High Medium Low
20. Frequency of update	Frequent Not frequent
21. Comprehensiveness of update	High Medium Low

Stage 1. Need for Purchasing Simulation Software

This stage represents the beginning of the software selection process. Once the need for purchasing simulation software has been established, several factors have to be considered. These factors include the intended simulation purpose, the existing constraints within the company, the main types of problems to be simulated, and information regarding the modellers and potential users. With regard to the intended simulation purpose, it should be decided whether the simulation software is going

Table 13. Criteria for pedigree.

Pedigree	
Criteria	Classification
1. Age	New Medium Old
2. Genealogy	- - - - - - - -
3. Spread	High Medium Low
4. Success	High Medium Low
5. Availability of references	High Medium Low
6. Software maturity	High Medium Low
7. Reputation of supplier	High Medium Low
8. Sources of information about the package	Literature Other users Supplier Demonstration Combination of several sources

to be used for education, "quick and dirty" modelling in industry or for complex/detailed modelling in industry, and/or research. If education is the intended software purpose, it should be determined whether teaching will be performed at the undergraduate or postgraduate level.

Other issues to be considered are organisational constraints. Financial constraints might include the hardware available for use by the simulation software, and the budget available for software purchasing, installation and maintenance costs, purchasing additional hardware, training of personnel, etc. Another constraint is the time available for software evaluation, selection and implementation.

A determination of the types of models that are likely to be developed can further help reduce the list of possible software candidates for evaluation. Three main types of model can be distinguished: discrete-event, continuous, or those that combine discrete and continuous elements. The issues relating to the persons involved in software selection, modelling, and the use of future models must also be addressed. Preferably, the same employees should be involved in the process of software selection and modelling.

Stage 2. Initial Software Survey

Following a determination of the above preliminary elements in Stage 1, an initial software survey has to be undertaken. The purpose of this is to shorten the list of software products that can be considered for evaluation and subsequent selection. If the initial elements are determined, the range of choice should already be narrowed. For example, if the systems to be simulated comprise both discrete and continuous elements, then all packages that are not suitable (i.e. cannot simulate both types of element) can be eliminated. At this stage, several other sources of information have to be consulted. Vendors of software products that seem to be candidates for software evaluation should be contacted and asked for assistance. They should provide as much information as possible, in addition to software demonstrations and written material. Other sources of literature related to software being considered ought to be examined and other software users contacted, if possible. The final decision for choosing software for evaluation should be influenced by the willingness of vendors to provide software for a free trial (at least a simplified version of the software) and appropriate documentation. Documentation should be well organised, indexed and written for an average non-technical user. The outcome of this stage is a short list of simulation software for evaluation.

Stage 3. Evaluation

Once it is decided which software products are to be evaluated and selected from Stage 2, the actual process of evaluation is performed. The main elements for this process are the evaluation framework presented in this paper, software to be evaluated, and documentation. Evaluation criteria must be chosen and the appropriate hierarchy determined for these criteria.

The first step is to establish the most important software features according to their intended purpose, and after that investigate additional features within groups of criteria, according to available time and preferences. It is advisable to develop a preliminary model that is typical for the intended software purpose and type of system to be modelled. This practical work is necessary for evaluation, because it gives an impression about the software and actually tests software facilities (it is not impossible that certain features work on paper but not in practice). At a certain stage of evaluation, after some notes have been made and it is clear which features are important and which additional information is needed, it might be useful to meet a vendor representative. This ought to enable a discussion of present and forthcoming features of the particular software to take place with someone who has an adequate level of technical expertise.

Stage 4. Software Selection

The actions described in the previous three stages should lead to credible software evaluation results. On the basis of these results, it should be decided which software seems to be the most appropriate and which are suitable alternatives (if there are any). In an ideal situation, it should be possible to purchase more than one software product for a particular purpose. In this case, software tools should be chosen on the basis of evaluation results.

Stage 5. Negotiating Software Contract

Following software selection in Stage 4, the next step is to negotiate a software contract acceptable to both parties. The contract should specify what products and services are provided, where and when they may be used, how the licence may be transferred to other parties, and how long the product may be used. Dates and obligations should be specified precisely to avoid any future misunderstandings.

Stage 6. Software Purchase

In this stage, software is purchased and implemented if an acceptable contract is acquired in Stage 5. Where a suitable agreement cannot be achieved, nor an adequate level of support secured, the subsequent best alternative should be examined on the basis of evaluation results. This requires a return to Stage 4 for the selection of alternative software products, then proceeding to Stage 5 for the negotiation of another software contract, and finally to Stage 6 for software purchasing after the acceptable agreement has been achieved with the software supplier.

4.4 Case Studies

The evaluation framework for simulation software evaluation and simulation software selection methodology presented in this paper have been tested through several case studies. These case studies refer to the selection of simulation software used for education and research at the Department of Computer Science and Information Systems at Brunel University. Simulation software for education is used in the final year undergraduate and MSc courses, whereas research in simulation is carried out at the Centre for Applied Simulation Modelling (CASM) [15]. Various consulting assignments have been carried out within CASM, involving companies in manufacturing [16], health [17], oil, water treatment and other industries. These companies have been assisted, amongst other services, in the process of simulation software selection by using the framework presented in this paper. Some aspects of the presented criteria were used in selecting simulation software for CLINSIM [17] development. Furthermore, this framework formed a basis for the development of SimSelect [18], a system for simulation software selection developed in Visual Basic 3.0.

The application of the evaluation framework presented in this paper within CASM has resulted in the purchase of various simulation languages and simulators, which are now being successfully used both for education and for research. For example, software with a quick learning ability was selected for undergraduate teaching, whilst more robust and flexible software packages were chosen for research and consultancy. Software currently available at CASM for simulation includes: WITNESS, SIMFACTORY II.5, INSTRATA, FACTOR, AIM, ProModelPC, MicroSaint, MODSIM, Simul8, AutoMod II, Taylor II, SIMPLE ++, ARENA and SIMAN/Cinema. Others are being added.

5. Concluding Comments

This paper provides a number of criteria derived for the evaluation of simulation software. More than 230 evaluation criteria are available within several groups of criteria. It would not be realistic to expect a particular package to satisfy all criteria. Nevertheless, some indication of which criteria are more important, according to the software purpose, is given. For each particular case of evaluation, it has to be decided which particular software features are more important than others. For this purpose, many other factors should be considered prior to software evaluation, such as financial and organisational constraints, individual preferences and experience of potential software users, types of systems that are to be simulated, etc. The evaluation framework and simulation software selection methodology presented in this paper can be used by anybody wishing to evaluate and choose adequate simulation software. This approach can improve the chance of effective simulation software selection and implementation. A specification of the initial factors to be considered once a need for purchasing simulation software has been established, forms a basis for the initial software survey. This should enable instant elimination of many software products that would not be suitable for a particular circumstance, and subsequently result in a short list of software for evaluation. An overview of the desirable characteristics of simulation packages provided in this paper is more comprehensive than those reviews found in the literature. Therefore, these guidelines can be used both by users who are looking for a suitable simulator to buy, and by developers of such simulators to improve existing versions of simulators or perhaps to develop a new, better simulation package.

References

1. W. G. Wild and P. Otis, "This video game is saving manufacturing millions", Business Week, pp. 82–84, 17 August 1987.
2. J. Banks, "Software for simulation", in C. Alexopoulos, K. Kang, W. R. Lilegdon and D. Goldsmith (ed.), Proceedings of the WSC'95 – Winter Simulation Conference, Washington DC, USA, pp. 32–38, Association for Computing Machinery, New York, December 1995.
3. J. W. Rice, "Simulationn in manufacturing", Proceedings of APICS 31st Annual Conference, Las Vegas, USA, pp. 410–413, 1988.
4. S. Wilkinson, "Simulation gives clear picture of assembly line", PC Week, p. 36, 40, 2 February 1988.
5. A. K. Kochhar and X. Ma, "Discrete event simulation software tools for the simulation of advanced manufacturing systems", Proceedings of the 1989 European Simulation Conference, Italy, Simulation Computer Society, San Diego, pp. 13–18, June 1989.
6. S. W. Haider and J. Banks, "Simulation software products for analyzing manufacturing system", Industrial Engineer, pp. 98–103, 31 July 1986.
7. F. H. Grant, "Simulation in designing and scheduling manufacturing systems", in W. D. Compton (ed.), Design and Analysis of Integrated Manufacturing Systems, National Academy Press, Washington DC, pp. 134–147, 1988.
8. A. K. Kochhar, "Computer simulation of manufacturing systems – 3 Decades of Progress", Proceedings of the 3rd European Simulation Congress, Edinburgh, Simulation Computer Society, San Diego, pp. 3–9, 1989.
9. J. G. Bright and K. J. Johnson, "Whether VIM? – A developer's view", European Journal of Operational Research, 54, pp. 357–362, 1991.
10. J. Szymankiewicz, J. McDonald and K. Turner, Solving Business Problems by Simulation, McGraw Hill, London, 1988.
11. G. T. Mackulak and P. A. Savory, "Ascertaining important features for industrial simulation environments", Simulation, pp. 211–221, October 1994.

12. J. Banks, J. S. Carson and B. L. Nelson, Discrete-Event Systems Simulation, Prentice-Hall, New Jersey, 1996.
13. V. Hlupic and R. J. Paul, "A methodological approach to manufacturing simulation software selection", Computer Integrated Manufacturing Systems, 9(1), pp. 49–55, 1996.
14. V. Hlupic and R. J. Paul, "Selecting software for manufacturing simulation", Proceedings of the 15th International Conference "Information Technology Interfaces", Pula, Croatia, University of Zagreb Computing Centre – SRCE, Zagreb, pp. 387–342, June 1993.
15. R. J. Paul and V. Hlupic, "The computer aided simulation modelling environment revisited", in J. D. Tew, S. Manivannan, D. A. Sadowski and A. F. Seila (eds), Proceedings of the WSC'94 – Winter Simulation Conference, Orlando, USA, December 1994, Association for Computing Machinery, New York, pp. 641–648, 1994.
16. V. Hlupic and R. J. Paul, "Simulating an automated paint shop in the electronics industry", Simulation Practice and Theory, 1(5), pp. 195–205, 1994.
17. J. Kuljis and R. J. Paul, "Organizing outpatient clinics using simulation modelling", The International Journal of Management and Systems, 10(30), pp. 299–306, 1994.
18. Hlupic, V. and A. S. Mann, "SimSelect: a system for simulation software selection", in C. Alexopoulos, K. Kang, W. R. Lilegdon and D. Goldsmith (eds), Proceedings of the WSC'95 – Winter Simulation Conference, Washington DC, USA, Association for Computing Machinery, New York, pp. 720–727, December 1995.