Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

HCI AND SIMULATION PACKAGES

Jasna Kuljis

Department of Mathematical and Computing Sciences
Goldsmiths, University of London
New Cross, London SE14 6NN
UNITED KINGDOM

ABSTRACT

Computer-based simulation modelling is one of the domains that is particularly demanding in terms of user interfaces. Issues that influence the 'usability' of such systems are examined. Several representative systems were investigated in order to generate some general assumptions with respect to those characteristics of user interfaces employed in simulation systems. There is a need for simulation systems that can support the developments of simulation models in many domains, which are not supported by contemporary simulation software. Many user interface deficiencies are discovered and reported. Proposals are made on how user interfaces for simulation systems can be enhanced to match better the needs specific to the domain of simulation modelling, and on how better to support users in simulation model developments.

1 WHAT IS HCI?

To users, the interface is the system (Hix and Hartson, 1993). Computer systems often lack good user interfaces for a variety of reasons, including the lack of a good user interface design methodology and the lack of good tools to implement a user interface. An interface is often the single most important factor in determining the success of a system (Larson, 1992). It is also one of the most expensive. Smith and Mosier (1984) conducted a survey of people concerned with information systems design who on average estimated that 30 to 35 percent of operational software is required to support the user interface. Bobrow et al. (1986) claim that the user interface often constitutes one third to one half of the code of typical knowledge-based systems. This claim is reinforced by Myers and Rosson (1992) who argue that anywhere from an average of 48% to a maximum of nearly 100% of the code for an interactive system is now used to support the user interface.

There is no agreed upon definition of the range of topics which form the area of human-computer interaction. The following definition is one from ACM SIGCHI (1992): "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them". Benyon and Murray (1988) make an important distinction between the terms Human-Computer *Interaction* and Human-Computer *Interface*. Interaction includes all aspects of the environment such as the working practices, office layout, provision of help and guidance, and so on. The interface is the part of a system with which the user comes into contact physically, perceptually or cognitively. We are mostly concerned with the interface part of simulation systems.

In this paper we examine the usability and appropriateness of such approaches when dealing with simulation software development. We aim to examine user interfaces for discrete event simulation packages. In particular, to investigate issues that influence 'usability' of simulation systems. Usability has many meanings to many different people. Software vendors often claim that their products have attributes such as high level of software usability, or 'user friendliness'. However, this terminology mostly indicates that software has Windows like GUI. There is no generally agreed definition of usability. A definition proposed by the International Standards Organization (ISO) and listed in Booth (1989) states: "The usability of a product is the degree to which specific users can achieve specific goals within a particular environment; effectively, efficiently, comfortably, and in an acceptable manner." A more operational definition is given by Shackel (1991) who suggests that any system should have to pass the usability criteria of effectiveness, learnability, flexibility, and user attitude.

Shackel's four distinguishable and quantifiable dimensions effectiveness, learnability, flexibility, and attitude are not mutually exclusive in the sense that measures of, for example, effectiveness can, at the same time, also give some indication of system learnability. Effectiveness refers to levels of user performance, measured in terms of speed and/ or accuracy, in terms of proportion of task(s), proportion of users, or probability of completion of a given task. Flexibility refers to variations in task completion strategies supported by a system. Learnability refers to the ease with which new or occasional users may accomplish certain tasks. Attitude refers to user acceptability of the system in question. There are some problems, however, in setting appropriate numerical values against usability goals. This approach would be more appropriate when the usability goals are set during the design stage of requirements specification than as an evaluation criteria once a system is finished. Although quantitative data is required to accurately assess the usability of a system, it is qualitative information that informs designers how to change an unusable system. Our objectives are both to assess the usability of current simulation systems and to identify usability defects.

The usability evaluation of the representative simulation system was carried out using structured walkthrough (Booth, 1989), i.e. we worked through a series of tasks the user might be expected to perform looking for sources of potential difficulties. We have based the examination on simulation software for personal computers partly because the issues of interaction are not dependent on the computer platform and partly because the PC platform predominates among commercial simulation systems. Therefore, the results and findings can be generalised across the whole spectrum of simulation software regardless of the host system.

Table 1: Main System Characteristics

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Application area	Manufacturing	Mainly manufacturing	Mainly manufacturing	General purpose	Mainly manufacturing	General purpose
Software type	Data-driven simulator	Data-driven simulator	Data-driven simulator	Data-driven simulator	Data-driven simulator	Language
Interaction style	Menus and fill-in forms	Menus and fill-in forms	Windows GUI	Windows GUI	Windows GUI	Windows GUI
Interaction devices	Keyboard	Keyboard and mouse	Keyboard and mouse	Keyboard and mouse	Keyboard and mouse	Keyboard and mouse
Navigation facilitated using	Function keys	Mouse, arrow keys, and function keys	Mouse, arrow keys, and F1	Mouse, arrow keys, and function keys	Mouse, arrow keys, and F1	Mouse, arrow keys, and F1
Terminology	Manufacturing	Manufacturing	Manufacturing	No specific domain	Manufacturing	No specific domain
Screen layout	Overcrowded	Good	Good	Good	Poor	Good
No. of colours	12	16	64	16	16	64
Use of colours	Hideous	Good	Good	Good	Too extensive	Good

2 HCI FEATURES IN EXAMINED PACKAGES

We have reviewed six discrete simulation systems: XCELL+, Taylor II, ProModel for Windows, Micro Saint for Windows, WITNESS for Windows, and Simscript II.5 for Windows. We examine the following interaction characteristics of these systems: input-output devices employed, interaction styles, and use of graphics. We are also interested in: type of simulation system, application areas, hardware platform, operating system, and hardware requirements. Table 1 provides a summary of the main characteristics of each system.

We test each of the six listed systems on the task of developing a small queuing model (a bank). The test is performed by a user with a high computer literacy, low domain knowledge, and no knowledge of any of the six simulation systems. The ability to accomplish the task was based solely on consulting the user manuals and the available on-line help (i.e. without formal training). We try to identify which of the general usability principles are applied and also establish where the usability defects are in each examined system. When examining the user interface we are particularly interested in three aspects: firstly, how the user interface for a particular system aids the user in a model development process; secondly, can the user modify the existing interface to either accommodate the user's own preferences or to adjust the modelling environment to the needs of a particular model; and thirdly, does the system facilitate user interface development.

2.1 Simulation Environments

The provision of completely self-sufficient simulation environment is important for the following reasons:

- it can reduce the development time,
- it can support application consistency,
- it can aid the developers throughout the development cycle,
- it can support model completeness,
- it can provide checks of model validation.

The experience we have gained during this research convinced us that the model development process is generally not well supported. Five of the six examined simulation packages are data-driven simulators (XCELL+, Taylor II, ProModel for Windows, Micro Saint, and Witness for Windows) that use some sort of diagramming tools for representation of model logic, and one is a simulation language (Simscript II.5 for Windows). All six systems provide modelling environments. Two systems are general purpose (Micro Saint, Simscript II.5) whereas the other four are manufacturing or mainly manufacturing. Graphic elements for the representation of the model logic are pre-defined for all simulators, and cannot be changed for two of them (XCELL+ and Micro Saint). Names of model elements (i.e. machines, parts) are pre-defined and cannot be changed for any of the simulators, although the user can provide labels for individual instances of elements to describe better the domain related elements (i.e. an element machine can be labelled 'clerk' or 'bank teller' in a bank model using Taylor II). Similarly, all examined simulators use fixed, pre-defined, and unmodifiable attribute names (fields). Data entry is usually facilitated through pre-defined, unmodifiable fill-in forms which use the system's own element names, attribute names (fields), which usually have default values provided. Data validation is not a common facility (available in Taylor II and ProModel for Windows).

Background drawing tools are rarely facilitated (Taylor II, ProModel for Windows, Simscript II.5 for Windows), as is importing graphics from other applications (ProModel for Windows, Micro Saint, Simscript II.5 for Windows). Icon editors are more common (not provided in XCELL+ and Micro Saint), even though the majority of them only provide elementary drawing capabilities. The user is rarely allowed to control statistics collection (only in Micro Saint and Simscript II.5) and the way the statistics is displayed (only Simscript II.5 gives complete freedom). Report customisation is rarely allowed. If this facility is provided, only a limited set of options can be exercised. On-line help, if available, usually does not extend to anything more than an overview of basic system concepts. Context sensitive help is scarce and good context-sensitive help is almost non-existent (the exception is ProModel for Windows). On-line help for error messages is not available on the examined systems. The customisation of the modelling environment is virtually an unknown commodity. A limited customisation is offered only in ProModel for Windows. The development of separable user interfaces for particular simulation problems is possible only in Simscript II.5 for Windows, which facilitates user interface development by providing templates for menus, fill-in forms, and several types of graphs that can be then tailored to suit the problem.

2.2 Data Input

It is apparent that data input part of the system is considered as less important than, for example, the visual simulation part. Most of the papers on simulation systems only briefly mention the data input capabilities of systems, if at all. However, there is room for a great deal of improvement in the domain of data input and/or model specification that would improve existing simulation systems. We have already mentioned that data validation is supported in only two of the six examined simulation systems. None of the systems offers database capabilities for keeping multiple variations of a model. Data input forms, if available, are generally poorly designed. There is no help provision for individual data fields. Importing data files is supported in four of the examined systems. The format of imported data is usually an ordinary ASCII text file. Therefore, there is much to be improved in the way the simulation data is communicated to the systems.

Table 2 summarise user interfaces for data input/model specification for the systems examined. Most of the systems keep data in text files that can be accessed and modified from other environments. None of the examined systems provides database facilities. Simulation languages, like SIMSCRIPT II.5 for example, can provide most of the data input features (menu-driven system, data-input forms, on-line help, data validation, etc.) but at the expense of an extensive time-consuming programming effort. Some of the systems provide limited input error checking and model verification facilities.

Table 2: Data Input/ Model Specification

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Model logic representation	Diagramming tools	Diagramming tools	Diagramming tools	Diagramming tools	Diagramming tools	Program
Graphic elements	Pre-defined cannot be changed	Pre-defined can be changed	Default or user selected	Pre-defined cannot be changed	Pre-defined can be changed	Not provided
Model elements	Pre-defined	Pre-defined	Pre-defined	Pre-defined	Pre-defined	User defined
Element names	Up to 10 characters	Up to 8 characters	Up to 80 characters	Up to 20 characters	Up to 8 characters	User defined
Attribute names	Pre-defined cannot be changed	Pre-defined cannot be changed	Pre-defined cannot be changed	Pre-defined cannot be changed	Pre-defined cannot be changed	User defined
Default values provided	Yes	Yes	Yes	No	Yes	Can be programmed
Fill-in forms design	Not applicable	Good	Good	Well balanced	Poor and often confusing	Not applicable
Importing files supported	No	Yes	Yes	No	Yes	Yes
Data validation supported	No	Yes	Yes	No	No	Can be programmed
Model validation supported	Partially	No	No	No	No	Can be programmed

2.3 Visual Simulation

Visual programming tools are standard features in all visual interactive simulation (VIS) systems, and drawing tools are very common. Dynamic icons and animation are supported by most visual simulation systems. The interactive change of the simulation parameters and of the speed of animation whilst the simulation is being executed, are also often provided. Panning and zooming is another quite common facility. Most of the current simulation systems have some form of visual animation of a simulation run. Animation speed in simulation systems is commonly made adjustable by their users. There are some problems with the animation speed that are not envisaged by the software developers. Simulation software is built for a particular hardware configuration and therefore for a particular processing speed (in MHz). The speed of animation (moving icons) is dependent on the computer processor speed. Hardware developments are much faster than software developments and by the time simulation software, based on a particular configuration, has reached the market it may well happen that the market has already adopted much faster computers. The user will probably install software on a much faster computer than it was intended for. Even though the user may have a facility to change animation speed, the slowest available speed may still be too fast for an animation observer. Table 3 provides a summary of some of the user interface features relevant to the design of simulation experiments.

Table 3: Simulation Experiment

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Background drawing tool	No	Yes	Yes	No	No	Yes
Icon editor	No	Yes	Yes	No	Limited capabilities	Yes
Importing graphics supported	No	No	Yes	Yes	No	Yes
Zooming supported	No	No	Yes	No	Using virtual windows	No
Panning supported	No	Limited (with arrow keys)	Yes	No	Limited	No
Interactive speed change	Yes	Yes	Yes	Yes	Limited to 3 speeds	Can be programmed
Interactive time change	Yes	Yes	Yes	No	No	Can be programmed
Interactive change of other simulation parameters	Yes	Yes	Yes	No	No	Can be programmed

Table 4: Simulation Results

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Control of statistics collection	System	System User partially	System User can only reduce the set	User	System	Modeller
Default statistics provided	Yes	Yes	Yes	Yes	Yes	NA
Graphics supported	No	Yes	Yes	Yes	Yes	Yes
Graph types	NA	Histogram, Pie chart Bar, Line, Scatter, Gantt, Area graph	Pie chart, Plot, Histogram, Bar, Line, Vertical line, and Step graph	Bar, Line, Scatter, and Step graph	Histogram Time series	Histogram, Pie chart, Line graphs, Dynamic bar graphs, Trace and X-Y plots, Dials, Meters
User defined presentation	No	Partially	No	Partially	Partially	Yes
Flexibility of presentation	None	Small	Small	Small	Small	Great
Tabular form statistics	Yes	Yes	Yes	Yes	Yes	Yes
Exporting files supported	Yes	Yes	Yes	Yes	Yes	Yes
Printing statistics tables	Yes	No	Yes	Yes	Yes	Yes
Printing graphs	NA	No	Yes	Yes	No	No

2.4 Simulation Results

Besides the standard reports for commonly occurring performance statistics (e.g. utilisation, queue sizes and delays, and throughput) some of the newer software allows the development of tailored reports. The user can choose the form of representation (e.g. textual output, table, representational graph, bar diagram, pie chart, histogram, time series), colours, labels, etc. In most of the VIS software, partial statistical results can be viewed during the simulation run. Graphs are updated dynamically during the simulation run. Some of the software has facilities to save the statistics into files that can then be printed. Rarely is there a facility to print the whole screen at any point of a simulation run or when the statistics are displayed/presented on the screen. Table 4 gives a summary of presentation of simulation results capabilities of the reviewed software.

Table 5: Printed Manuals

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Tutorial	Not provided	Not thorough enough	Not provided	Yes	Partially covered in the only manual.	Not provided
Getting started	Part of the User Guide	Not provided	Yes	Yes	Not provided	Not provided
User guide	Yes	Yes	Yes	Yes	Yes	Yes
Reference manual	Not provided	Not provided TLI reference manual provided	Yes	Not provided	Not provided	Yes
Index	Only XCELL+ glossary	Global for all manuals	System concepts	System concepts	System concepts	System concepts
Terminology	Manufacturing	Manufacturing Technical	Manufacturing	General simulation	Manufacturing	General simulation. Technical

2.5 User Support

As a rule documentation is highly erratic, written in a technical jargon and rarely organised in any structured manner. If it contains an index it usually requires the user to know the exact terminology used to be able to find the topic of interest. Similarly, the installation procedures are generally badly documented and off-putting All of the systems examined, except Simscript II.5, require a security. Considering the high cost of simulation software it makes some economic sense from the vendors point of view. However, from the customers point of view it shows a basic mistrust in customers' honesty and adds to the general unfriendliness of the simulation systems. User manuals are usually poorly written. Of the six simulation systems we have examined only two provide Tutorial (Taylor II and Micro Saint), three provide Getting Started (XCELL+, ProModel for Windows, and Micro Saint), two provide Reference manuals (Pro Model for Windows and Simscript II.5). An index is provided in all of them except XCELL+, but it usually lists only system concepts using a particular simulation system's terminology. Examples, if provided, are not followed throughout the development process and are therefore of not much use. The summary of characteristics of printed manuals is given in Table 5.

On-line help very rarely provides help for all facilities and the available tools in the simulation environment. Context-sensitive help is a rare commodity (it is only provided in Taylor II and ProModel for Windows) and is almost unheard of for system messages (i.e. error messages). An on-line tutorial is provided only in ProModel for Windows. Demonstration disks are provided for three of the examined systems Taylor II, ProModel for Windows, and Micro Saint, where ProModel provides a professional and carefully thought out product. Table 6 provides a summary of on-line user assistance offered in the six examined simulation packages.

3 USABILITY OF SIMULATION PACKAGES

Effectiveness of the system can be hindered if there are: defects in navigation through the system, problems in screen design and layout, inappropriate terminology, inappropriate feedback or complete lack of feedback, problems with modality, inconsequential redundancies, and problems in matching with user tasks. Learnability can be impeded if there are: defects in navigation, problems in screen design and layout, inappropriate terminology, inappropriate feedback or complete lack of feedback, and problems in matching with user tasks. Flexibility is impeded if there is no user control over the system and if the system imposes the order of the steps in a task. User attitudes towards the system can be seriously affected by any of the above usability defects.

The usability defects were identified in examined simulation systems. ProModel for Windows has only a few problems, which are its terminology and visual objects that are appropriate solely for the manufacturing domain. XCELL+, Witness for Windows, and Simscript II.5 have serious defects in navigation through the system. Feedback is inadequate in all five packages except in ProModel for Windows. Consistency of a system is assessed based on the degree to which the system performs in a predictable, well organised and standard fashion. XCELL+ has an inconsistent use of function keys, Taylor II an inconsistent use of interaction devices, and Simscript II.5 suffers from the inconsistency

Table 6: On-line User Assistance

	XCELL+	Taylor II	ProModel	Micro Saint	Witness	Simscript II.5
Model examples provided	Yes	Yes	Yes	Yes	Yes	Yes
Help type	One text screen	Isolated text screens	Hypertext	Limited hypertext	Isolated text screens	Hypertext
Navigation through help facilitated using	Not applicable	Mouse and arrow keys	Mouse point and click on link nodes.	Mouse point and click on link nodes.	Mouse point and click on cross reference buttons	Mouse point and click on link nodes.
Help text	Short information on system	Complete version of printed material	Differs from printed manuals	Differs from printed manuals	Differs slightly from printed manuals	Differs from printed manuals
Index of topics	Not applicable	Yes	Yes	Yes	Yes	Yes
Search facility within help	Not applicable	Searches for a first occurrence of a given string	Yes	Searches for a first occurrence of a given string	Not provided	Yes
Tutorial	Not provided	Not provided	Interactive lessons	Not provided	Not provided	Not provided
Context-sensitive help	Not supported	Only relevant page	Yes	Not provided	Not provided	Not provided
Help on help	Not provided	Not provided	Extensive	Limited	Not provided	Yes
Printing help text supported	No	No	Yes	Yes	No	Yes
Demonstration disk	Not provided	Elementary	Professional	Yes	Not provided	Not provided

across its system modules. Modality is the state of the system operation that the user selects to perform a particular function. Only ProModel for Windows and Micro Saint provide clear feedback on the mode that the system is currently in. The four other systems either use modal dialogue which require the user to respond before any action can be taken or do not provide a clear indication in which mode the system is in. Only ProModel for Windows gives the user the feeling of being in control.

There is no provision, or if there is it is marginal, for user interface customisation. Only ProModel for Windows provides a minimal customisation of modelling environment. There is a facility for guiding an inexperienced user through the necessary steps of model development. Experienced users can choose the order in which to perform the steps of tasks. There is the possibility to create new simulation systems for limited domains with a custom made interface appropriate for the model domain. These capabilities are currently limited to bespoke programming that requires a substantial development effort. Sometimes user interface development can be facilitated using an object-oriented approach that reduces development time.

Many authors argue that the advantages of VIS include better validation, increased credibility (and hence model acceptance), better communication between modeller and client, incorporation of the decision maker into the model via interaction, and learning via playing with the VIS. However, there is little published empirical evidence to substantiate these claims. In addition, animation can be used to enhance a model's credibility and, according to Law and Kelton (1991), it is the main reason for animation's expanding use. Swider et al. (1994) feel that animation can provide convincing evidence that model behaviour is representative of the system under study. Cyr (1992) see advantage of using animation in its ability to demonstrate problems with the model itself which would otherwise be difficult to detect. Kalski and Davis (1991) point out that summary statistics sometimes do not show the active interactions of processes in a system, and they advocate the use of animation as an aid to the analyst in identifying the system status.

There are many animation proponents in the simulation community, especially the software vendors. However, there is very little published empirical evidence which would suggest how to design effective animation. Carpenter et al. (1993) conducted an experiment with 47 subjects to examine how well the animation communicated the operation of the simulation model. They considered combinations of three aspects of animation - movement, detail of icons, and colour. The results suggested that movement of icons is more important than their detail or colour in communicating the behaviour of a simulation model with moving entities. Swider et al. (1994) used 54 subjects to obtain objective and subjective measures in determining which combinations of animation presentation and speed were best for displaying violations of model assumptions. Based on the results of this study, Swider at el. (1994) recommend: the use of pictorial display with moving icons for simulation models with moving entities; the facility to set the presentation speed to make discrete differences visible; and to avoid overloading the user with too much visual information.

The results of the above two studies are not surprising and they match our intuition and common-sense. However, their importance is in substantiating our intuitive judgement with some more concrete evidence. Animations with moving icons are often used in current simulation systems even though presentation of animation is not often well thought about. Ideally, it may seem desirable to present information on the screen that has characteristics similar to the objects we perceive in the environment. The visual system could then use the same processes that it uses when perceiving objects in the environment. Factors that contribute towards the meaningfulness of a stimulus are the familiarity of an item and its associated imagery. The graphical representation of constructs for different applications should give definite information about the type of model component it represents, such as waiting queues, customers or servers in queuing systems or stores, or suppliers in store keeping systems (Kämper, 1993). Stasko's (1993) animation design recommendations state that animation should provide a sense of context, locality, and the relationship between and after states. Furthermore that the objects involved in an animation should depict application entities and that the animation actions should represent the user's mental model. If these recommendations were followed, the effectiveness, learnability, and the enthusiasm of a wider user population to use simulation systems might increase.

The eye-catching, appealing nature of animation can tempt designers to apply too many facets to an interface. Animation is, however, another attribute in which the often quoted design principle "less is more" does apply. Nevertheless, if the screen design is kept clean, simple, and well organised some redundant information can be quite useful to the user. The moderation principle is something that many simulation system developers should learn about. User interfaces that have screens crowded with too many objects, large numbers of offensive colours and incompatible colour schemes is more of a rule than an exception.

An essential aid in model development can be facilitated by selecting model components which are relevant to the model builder's modelling requirements. Our proposal (Kuljis 1994, 1995) is that simulation environments should provide model developers with the following:

i) Several pre-defined problem domains.
ii) A facility to create new problem domains.
iii) A facility to design and/or choose graphical representations for elements in a problem domain.
iv) A facility to set default values for a problem domain.
v) A facility to set defaults for statistical data collection.
vi) A facility to set defaults for the graphical presentation of simulation results.

REFERENCES

ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group. 1992. *ACM SIGCHI Curricula for human-computer interaction.* Technical report, New York: Association for Computing Machinery, Inc.

Benyon, D. and D. Murray. 1988. Experience with adaptive interfaces. *The Computer Journal* 31(5): 465.

Bobrow, D.G., S. Mittal and M.J. Stefik. 1986. Expert systems: Perils and promise. *Communications of the ACM* 29(9): 880-894.

Booth, P. 1989. *An introduction to human-computer interaction.* Hove: Lawrence Erlbaum Associates.

Carpenter, M.L., Bauer Jr., K.W., Schuppe, T. F. and Vidulich, M. V. 1993. Animation: What's essential for effective communication of military simulation model operation? In *Proceedings of the 1993 Winter Simulation Conference*, 1081-1088. Institute of Electrical and Electronics Engineers, Los Angeles.

Cyr, R.W. 1992. Using animation to enhance a marine-terminal Monte Carlo simulaton. In *Proceedings of the 1992 Winter Simulation Conference,* 1000-1003. Institute of Electrical and Electronic Engineers, Arlington, Virginia.

Hix, D. and R.H. Hartson. 1993. *Developing user interfaces. Ensuring usability through product & process.* New York: John Wiley & Sons, Inc.

Kalski, D. R. and Davis, A. A. 1991. Computer animation with CINEMA. In *Proceedings of the 1991 Winter Simulation Conference,* 122-127. Institute of Electrical and Electronic Engineers, Phoenix, Arizona.

Kämper, S. 1993. A systematization of requirements for the user interface of simulation tools. *SAMS* 11 (2): 107-119.

Kuljis, J. 1994. User interfaces and discrete event simulation models. *Journal of Simulation Practice and Theory* 1(5): 207-221.

Kuljis, J. 1995. Usability of manufacturing simulation software. *International Journal of Manufacturing System Design* 2(2): 105-120.

Larson, J.A. 1992. *Interactive software. Tools for building interactive user interfaces.* Englewood Cliffs: Prentice Hall.

Law, A.M. and D.W. Kelton. 1991. *Simulation modeling and analysis.* 2nd ed., New York: McGraw-Hill

Myers, B. A. and M. B. Rosson. 1992. Survey on user interface programming. In *Proceedings of HCI Conference on Human Factors in Computing Systems,* 195-202. New York: ACM.

Shackel, B. 1991. Usability - context, framework, definition, design and evolution. In *Human Factors for Informatics Usability*, eds. Shackel, B. and S. Richardson. Cambridge: Cambridge University Press.

Smith, S.L. and J.N. Mosier. 1984. The use interface to computer-based information systems: A survey of current software design practice. *Behaviour and Information Technology* 3(3): 195-203.

Stasko, J.T. 1993. Animation in user interfaces: Principles and techniques. In *User Interface Software*, Bass, eds. L. and P. Dewan. Chichester: John Wiley & Sons, pp. 81-101.

Swider, C. L., Bauer Jr., K. W. and Schuppe, T. F. 1994. The effective use of animation in simulation model validation, *Proceedings of the 1994 Winter Simulation Conference,* 633-640. Institute of Electrical and Electronic Engineers, Orlando, Florida.

AUTHOR BIOGRAPHY

JASNA KULJIS is a Lecturer in the Department of Mathematical and Computing Sciences at Goldsmiths, University of London. She has a B.Sc. in Mathematics from Zagreb University, a M.S in Information Science from Pittsburgh University and a PhD in Information Systems from the London School of Economics. She is currently researching into human-computer interfaces to modelling systems, having previously spent over a decade as a researcher, consultant and teacher at the University of Zagreb Computer Centre and seven years as a Lecturer at the Roehampton Institute London. She has published widely in many aspects of computing, and has undertaken consultancy for a variety of government agencies and companies in Croatia, as well as for the Department of Health in the U.K.