Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

MODELLING STYLES AND THEIR SUPPORT IN THE CASM ENVIRONMENT

DAVID W. BALMER
Department of Statistical and Mathematical Sciences
London School of Economics and Political Science
Houghton St.,London WC2A 2AE, England.

ABSTRACT

Much effort is currently being expended in the design and implementation of integrated software support environments for discrete event simulation modelling. What are the considerations underlying the development of such environments and how are efforts being directed? This paper discusses aspects of the background to current developments and relates this discussion to the context of one such environment namely that of the Computer Aided Simulation Modelling (CASM) group at the London School of Economics.

1. INTRODUCTION

The general concept of an integrated software support environment (ISSE) for discrete event simulation modelling has been gaining wider acceptance with many such environments being designed or under construction (Nance 1979; Henriksen 1983; Reese and Sheppard 1983; Standridge and Walker 1983; Smart and Baker 1984; Rozenblit and Zeigler 1985; Balci 1986; Balmer and Paul 1986; Lehman et al. 1986; Nance 1987). All proponents of such environments no doubt share the common intention that these should improve the productivity (and creativity) of simulation modellers as well as extending the application and effectiveness of simulation techniques. Despite certain areas of broad agreement about the general functional components of an ISSE, important differences of view remain concerning the detail, structure and direction of development.

In the face of the variety of modelling context and present state of knowledge of the 'art of modelling', this plurality must be seen as healthy. However, this very variety of context and the range of differing perspectives and intentions of the developers of ISSEs make the comparison and assessment of alternative offerings exceedingly difficult. Such evaluation, whether formal or informal, is important not only for the intending 'customer' looking for an appropriate ISSE, or the system developer seeking new inspiration but also that these developments should make their proper contribution to the growth of understanding of modelling processes.

The remainder of the paper is divided into two main sections followed by a brief summary and conclusions. The first of these sections contains a discussion of the general issues underlying the development of ISSEs and establishes a preliminary analysis of the development context. The second main section describes the evolution of the CASM support environment under five principal subheadings expanding on the account in Balmer and Paul (1986).

2. DEVELOPMENT OF SOFTWARE SUPPORT SYSTEMS

Figures 1 and 2 attempt to catalogue the dimensions of the ISSE development context and perceived modelling context, respectively. Despite the fact that the catalogues are obviously incomplete and some dimensions identified clearly demand a greater refinement, the lists are too lengthy to discuss each separate entry in detail. The discussion below focusses on selected items for further comment.

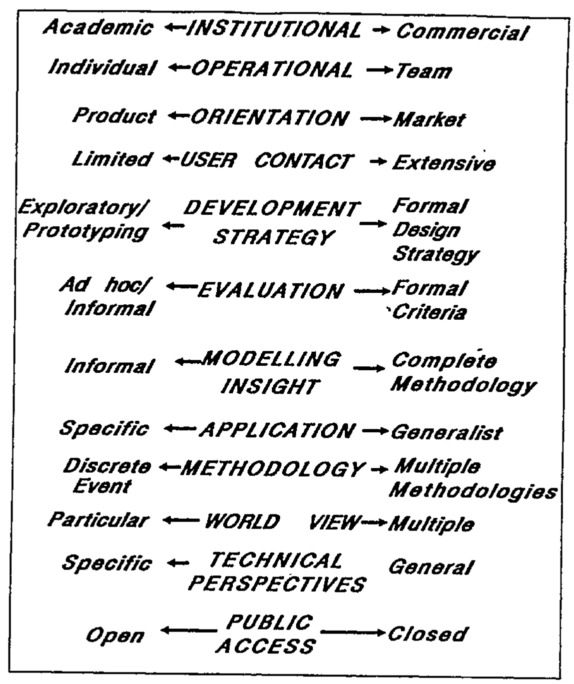

FIGURE 1. Dimensions of the ISSE Development Context

2.1 Development Context

Most support system proposals and implementations emanate from either professional software houses with a prior commitment to simulation software or academic groups with research interests in the modelling and simulation process. In either case personal and corporate experience of modelling and modelling requirements are prime determinants of the nature and form of the ISSE developed. Such experience might be supplemented by reference to standard literature on simulation modelling, to published surveys of simulation practitioners such as that of Christy and Watson (1983) or by the performance of specific market research amongst fellow modellers/clients or wider to define support requirements. In some cases the design and development of an ISSE might even enjoy the full benefit of close and continuing collaboration between system developers and representative user or group of users.

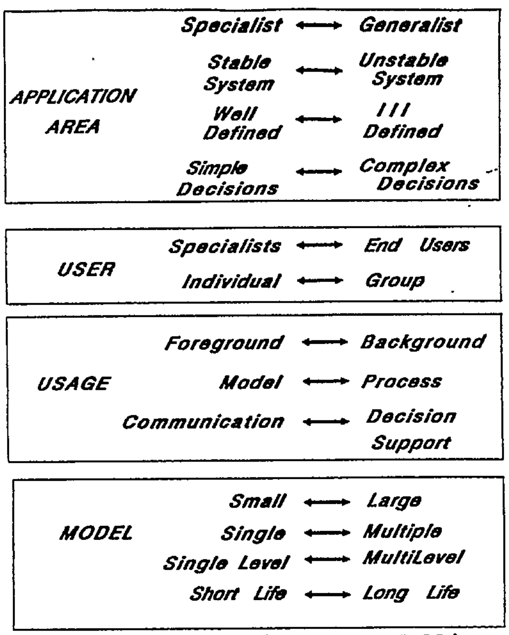

FIGURE 2. Dimensions of Modelling Perspectives

An ISSE implies a large-scale software development process which may proceed by carefully staged implementation of a comprehensive design or by rather piece-meal development of tools within a loosely integrated framework guided only by a broadly coherent perspective on modelling support requirements. The development time scan of a large system may be correspondingly long and the development strategy must either anticipate or embody a flexibility sufficient to accommodate advances in hardware and software technologies, in understanding of the 'art of modelling' and in user expectations. Current developments in IKBS not only offer the developer the use of advanced software technologies but may also encourage heightened user expectations of the level and character of support which ISSE developers may have to satisfy.

Full details of ISSE design and implementation, of the overall system and of intermediate stages, may be published and subjected to the critical appraisal of modellers and other system developers or may, perhaps for reasons of commercial interest, be withheld.

The evaluation of ISSE developments is important. An ISSE is the conscious (or unconscious) expression of a perspective of the modelling process expressed in terms of a particular computer implementation (Balci 1986). The success (or otherwise) of this implementation in supporting modelling activities will provide evidence of the value of the underlying perspectives. Assessment is vital to the development of a more comprehensive understanding of the arts of modelling. If the ISSE has been developed commercially or under contract then the evaluation is possibly to be found in the level of sales, in measures of client satisfaction or, at least, in the meeting of contractual commitments.

For the academic developer the question of evaluation is less likely to be answered in such terms. The proof of the efficacy of a modelling environment must be sought in its application. For the academic group this application will usually be either personal to the development group or by students, often those taught by that same group, rather than experienced modellers. Evaluation of this character is not worthless and will certainly lead to modifications and marginal enhancements but it is not without its difficulties. The education of future modellers is no trivial task (Soloman 1980) and an ISSE satisfying such pedagogic requirements may not be suitable for use by experienced modellers in real world application (Mathewson 1980).

2.2 Modelling Perspectives

Perhaps the most critical aspect of the ISSE development context is the view taken of the modelling process. The simulation literature contains accounts of the modelling process in standard texts or in tutorial papers such as Nance (1983) or Schruben (1983). The process is usually described as a series of steps possibly including some cycling. The descriptions are suggestive of a highly structured approach which seems uncharacteristic of much modelling practice. In a slightly different context Pidd and Woolley (1980) observed that the highly formalised approach to problem structuring found in the literature '...was not observed in the O.R. groups we visited'.

Schruben (1983) reviews a number of such descriptions and provides a tabular summary. One can recognise common emphases on 'formulate the problem', 'build the model' and 'analyse the model'. Whilst such steps may provide a broad framework they permit considerable variation in interpretation. Since the building of a simulation model will usually involve the construction of a large and complex computer program it may seem natural to seek parallels between software engineering practices and simulation modelling and to base at least some aspects of ISSE on such parallels. As indicated by Schruben's list there are other areas falling outside the attentions of software engineers where the provision of support is at least as important and the ISSE developer must look elsewhere for guidance.

Such further insights must be conditioned at least in part by the view of the modelling context. Whilst it may be true that some aspects of discrete event simulation modelling can be the subject of a general treatment, application context should have a strong influence on the precise forms and emphasis of support requirements.

Models are built for a variety of purposes and in a variety of contexts. Purposes may include the communication of facts or ideas about a system, the generation of new ideas for the design or operation of the system, prediction of system behaviour provision of insight and the support of decision making. They may be built by a single individual knowledgable about both system modelled and modelling techniques or in a group setting with each member contributing different complementary and conflicting influences and knowledge under one of many socio-economic frames. Perhaps the most common of such is the consultant analyst and client frame. Both purpose and context must influence form. For instance, a consultant might see an ISSE as part of the 'tools of the trade'and might welcome some technical mystery. A lone non-expert user may demand greater specificity and higher levels of support at the expense of flexibility and generality.

The system modelled may have components which are physical, mechanical or electrical, whose behaviour, although complex, is stable and subject to known physical laws. Other components may involve predominantly interactions of human participants whose behaviour may present complexities of a less stable sort. Human organisations are typically purposeful and adaptive and the modelling of such systems has to accommodate the fact that the system itself is usually changing rapidly. Such systems are typical of the province of O.R. modelling where problem solutions, model-based or otherwise, have but a short life (Ackoff 1979, Tobin et al. 1980). The pace of model development and use will accordingly have to match that of the decision-maker's environment.

The flexibilities required of modelling within such systems will be different in order and kind than those required within predominantly physical systems. The support of the acquisition of and representation of knowledge about the human aspects of systems will require specific facilities. Greater consideration must be given to the accommodation of the full range and complexity of management control options. Such a system would necessitate an 'exploratory approach' as described by Pidd and Woolley (1980) in the context of problem structuring. The rigidities of an overly formalised approach would certainly lead to the difficulties pointed out by by Rosenhead (1980) in his discussion of the requirement of 'robustness' in planning.

Some descriptions of the modelling and simulation process infer that the product of the process is a model or a decision support system based on the model developed during the process. Ackoff (1979), writing on modelling in the O.R. context, has stressed the process of modelling and the insights so generated as the principle product. This view likens the planning or modelling process to the design of an idealised system and the investigation and implementation in consequent decision-making of steps towards the ideal. The implication of this view is that model development and experimentation should be a continuing process involving the ongoing attention and participation of all those with an interest in the system, the 'stake holders'. The role of the modeller and similarly of an ISSE in this context is thus one of facilitator enabling the more adequate participation of stake holders in the process.

2.3 Issues of Style

The title of the paper uses the expression 'modelling styles' deliberately to emphasise that, in addition to the aspects of context and modelling domain, there are features of modelling that simply reflect the preferences of the human participants in the process. Bronowski (1973) in his famous reflections on the 'Ascent of Man' observes that

> 'The most powerful drive in the ascent of man is his pleasure in his own skill. He loves to do what he does well and, having done it well, he loves to do it better.'

Ackoff (1979), in dissecting the 'corpse' of O.R., discussed the notion of style. He identified style with the satisfaction to gained from the manner of performing tasks and accused O.R. people of neglecting, in their analyses, these satisfactions of 'means' in favour of those 'ends'. Most of the accounts of the simulation modelling process neglect all mention of the intense satisfactions to be gained from the process itself. A proper regard for these satisfactions is imperative if the full productive and creative potential of the participants is to be attained. The concept and form of an ISSE must, of course, support the activities of the modeller but neither should make the job of the modeller so automatic as to reduce its interest and challenge without the risk that modellers will resist its use. The modelling solutions produced within such an environment similarly must not reduce the role of decision-maker but must enhance it. Models are tools for thought and the modelling process supported by an ISSE must match the implicit mental models of participants and enlarge them.

A further set of satisfactions which it is well not to neglect in the consideration of the development context of ISSEs are those of the developers. The returns, in an academic situation, of such work are in no small part seen in terms of the technical and intellectual challenge presented by the task. Work on ISSEs is interesting, challenging and enjoyable and the design and forms of ISSEs undoubtedly reflect this.

3. APPLICATION TO CASM

3.1 The CASM Context

The CASM group consists of members of faculty and research students at the London School of Economics. It is primarily motivated towards research and teaching but enjoys various consultancy links with industry and government. The group's aspirations were outlined in Balmer and Paul (1986). They concentrate largely in the investigation of computer support for discrete event simulation modelling and are, in the main, exercised in a general context with neither commercial intent nor a collaborative sponsor.

The perceptions of modelling context underlying ISSE developments lean strongly towards operational research with a concern for the support of the 'process' as well as the 'model' concept of modelling 'product'. The expectation of a rapidly changing 'messy' modelling context leads to an emphasis on rapid, interactive model development and investigation and modular model structures permitting ready extension and modification.

Perspectives on the modelling process within the CASM group reflect varied individual experience and insight but broadly speaking focus on a familiar series of modelling phases with an emphasis on the lightly structured iteration of the basic process in an exploratory spirit.

The software development strategy is based on extensive exploratory prototyping of tools in a loosely integrated environment. Design details are published whereever possible and subjected to external appraisal. Evaluation draws on personal experience of application in appropriate consultancy contexts (Balmer and Paul 1985; El Sheikh et al. 1987). However, the evaluation relies heavily on performance of systems in various teaching contexts. Systems have been used by the group and others in a variety of university and polytechnic teaching environments in courses given to undergraduates and postgraduates with apparent success. The systems have also been used in connection with short courses for modellers of varying degrees of experience from within the scientific civil service and industry in the United Kingdom and elsewhere and have been well received.

3.2 Basic Computing Environment

<u>Minimal Support System.</u> The basic environment is centred around a simulation system consisting of libraries of declarations, functions and procedures and a program template in Pascal providing a structured framework for the development of simulation models. Other features of a minimal support environment such as file handling facilities, editor, compiler, linker, debugger are accepted from the host environment. For instance, a version used extensively for teaching is implemented in a PC hardware environment in TURBO Pascal. The TURBO Pascal system provides basic editing, compiling and debugging support. The basic system exists in a variety of mini and microcomputing environments. It offers the advantages of structure, modularity and focused conciseness, with the flexibility and support of a widely used high level language such as Pascal.

A complete account of this basic software environment can be found in Crookes et al. (1986).

The world view embodied in the system is that of 'three phase modelling'. The approach and its merits are discussed in Crookes (1982), Crookes et al. (1986) and OKeefe (1986). The arguments of the superiority of the three phase method are very compelling. It seems to offer clear advantages in the areas of modelling clarity, model maintainability and modularity.

<u>Extra Basic Support.</u> The minimal support system described above functions as a well proven teaching environment offering appropriate levels of user satisfaction. The principal disadvantage of working in a

```
Queue Name: Clean_Glasses

PROCEDURE B6 ;   (* End Wash *)
BEGIN
    ADDTO(Back ,          CURRENT);
END;
```

FIGURE 3. Local Code Generation within TURBO Pascal Editor

general high level language such as Pascal, supplemented by libraries is that all native support facilities such as debuggers are geared to the base language and offer little or no support in respect of the specialist library functions. This deficiency can only be remedied by the additional specific debugging support. Doukidis and Paul (1986) give details of an expert system, SIPDES, developed to diagnose, locate and suggest possible solutions to perceived 'errors' in simulation programs. These 'errors' may be reflected in compilation, execution or verification failures and a consultation with SIPDES starts with the user identifying the general area of the problem. From this point SIPDES leads the user through an appropriate series of enquiries until the fault is diagnosed.

The editor provided by the host environment is unlikely to have any features of particular relevance to its present role in model development. Balmer (1987a) describes a utility which has been developed to provide additional editor facilities within the TURBO Pascal environment. These extra facilities are accessible through single <alt> keystrokes and include specific online help tailored to the simulation system and localised automatic programming through a 'form filling' generation of program structures such as B and C Procedures for insertion in the developing program within the TURBO editor. Figure 3 shows a typical user screen invoked by '<alt>B'.

3.3 Automatic Programming.

Prototype Program Generator. However, the coding of a simulation model soon becomes a matter of dull routine and a facility for the automatic generation of complete computer code is a vital aspect of any ISSE. An Interactive Simulation Program Generator, (ISPG) is a software tool which requires the modeller to enter the formal specification of a simulation model. From such information, the ISPG produces a source program which is a computer-executable version of the model specified.

Chew et al. (1985) give details the prototype ISPG, LANGEN, developed within the CASM project. Paul and Chew (1987) describe its immediate successor, AUTOSIM. Both accept a model specification in terms of an Activity Cycle Diagram (ACD) and from this produce the source code of a Pascal simulation program supported by the basic system described above. The description of the ACD is communicated to each through a similar interrogative dialogue which, in external form and concept, owes much to the pioneering work of Clementson (1982).

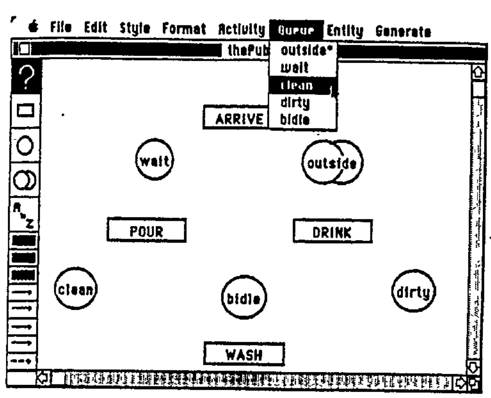

FIGURE 4. Graphical Interface for Model Development

Alternative Interfaces. The intrinsic strengths and limitations of either LANGEN or AUTOSIM lie in the power or weakness of the model specification language, the ACD, and the suitability of the particular form of user interface. The ACD is a simple, easy to understand formalism which is amazingly powerful and yet it has significant limitations (Balmer and Paul 1985). The interrogative interface of LANGEN or AUTOSIM does not encourage flexible model entry nor extensive model editing. However, the piece-meal, evolutionary development strategy of the CASM group promotes the simultaneous investigation of both alternative model definition languages and interfaces, described in detail elsewhere, each reflecting its own trade-off between power and ease of use (Paul 1986; Balmer and Mireskandari 1986; Paul and Chew 1987). Figure 4 illustrates the ACD-based graphics interface developed by Au (1987).

3.4 Extended Support System

Graphical Modelling Support. Mention has already been made of the development of graphic input facilities. The iconic

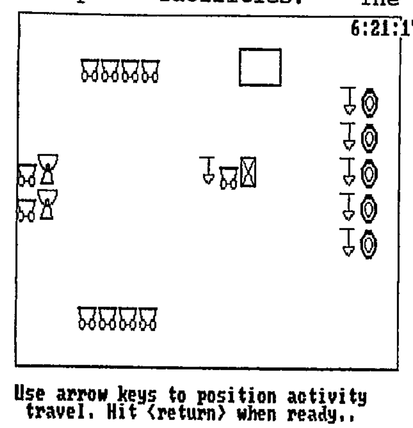

FIGURE 5. Animation for Simulation through AUTOGRAF

representation of a simulation model in execution offers improvements in participation in modelling processes and increased end-user acceptability of any models produced. Animation details may be added to the model description file produced by AUTOSIM through a graphics editor, AUTOGRAF, (Paul and Knox 1987). A simulation program including the support of animation may then be generated by AUTOSIM. Figure 5

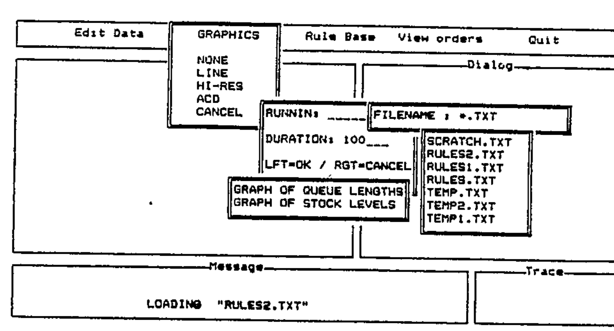

FIGURE 6. Interfaces for Simulation Models

shows some typical screens being produced.

Interface maker. A robust, attractive user interface is imperative if the co-operative attention of a busy manager is to be retained during the model development. Borrowing concepts from the best of modern software packaging, facilities have been developed whereby the modeller may design and add a multi-window screen, menu-driven interface with subsidiary pull-down menus under cursor key or mouse control to a basic simulation model. More details are given in Goodman and Balmer (1987) but Figure 6. shows a typical interface so produced.

3.5 Support of a Modelling Methodology

Some proponents of ISSEs base their proposals on more formalised analysis of the modelling process than has been the case within the CASM group (Zeigler 1984; Zeigler and de Wael 1986; Nance 1979, 1983, 1987).

The support of hierarchical modelling (HM) described in Balmer (1987b and 1987c), perhaps, comes the closest to the support of a formal modelling methodology. The process follows the basic form illustrated in Figure 7.

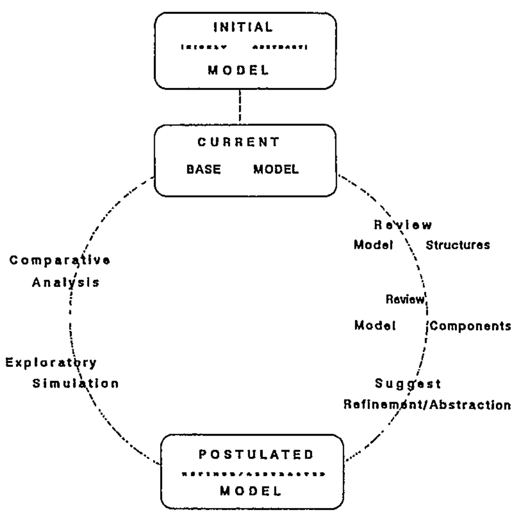

FIGURE 7. Hierarchical Modelling Process.

Model refinement (and abstraction) within an activity-based hierarchy may be guided by simple questionning but the suggestion and acceptance of modelling alternatives will remain a function of the modeller's skill and experience.

HM has a minimal support requirement for an ISSE allowing the description and the editing, storage and retrieval of model descriptions in a form reflecting their hierarchical structure and facilities for the comparison, analysis and exploratory simulation of models and sub-models. The loose iteration through these stages must be directed by a comfortable mixture of (artificially) intelligent machine guidance and user control.

The general structure of such a support environment is defined in Figure 8.

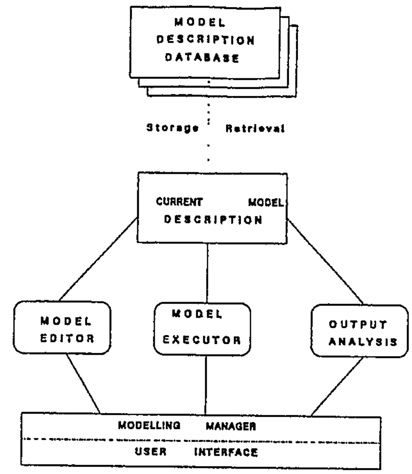

FIGURE 8. General Features of Software Support for Hierarchical Modelling

Figure 4 showed a tentative graphics facility for model specification. The obvious progression from an existing rudimentary ISSE for HM described in Balmer (1987b) is to develop and extend the graphics interface to allow the definition of activity-based hierarchical models. This could include detail hiding and recovery through 'zoom in' and 'zoom out' facilities.

3.6 Simulation and Artificial Intelligence

CASM developments have included AI based advisory systems including support for the earlier stages of model formulation has been developed in the form of both a rule-based expert system and a natural language understanding system and the fault diagnosis system mentioned earlier (Doukidis and Paul 1985,1986a, 1986b).

Interfacing Simulation/Expert Systems. Where the modelling context implies the need for the flexible representation of complex management control options, standard simulation modelling structures may prove inadequate. In this circumstance there would seem to be merit in separating the more formal and stable aspects of the model from the more emphemeral aspects of management control. The former would be represented in the familiar simulation structures, the latter in the form of an expert system and the two components suitably interfaced.

The three phase approach accommodates such decomposition particularly well. The C phase essentially involves the testing of the current model state against a set of rules which will determine which, if any, new activity starts are implied. In the face of complex, interrelated and evolving rule systems, such a search might best by effected within a rule-based expert system. The simulation executive looks to the associated expert system to deliver the management control decisions and the current state of the simulation model provides the context within which the expert system, with access all simulation state variables.

applies its knowledge.

Goodman et al.(1987) describe developments of this type within the CASM environment in the context of the simulation of an engineering job shop where the complex scheduling of production relies on the experience and judgement of the production controller.

Figure 9 shows the overall structure of the present system. It includes a simulation model which progresses the job shop operations through time, an expert system which manages the scheduling of tasks within the simulated system, associated data files and user interface system.

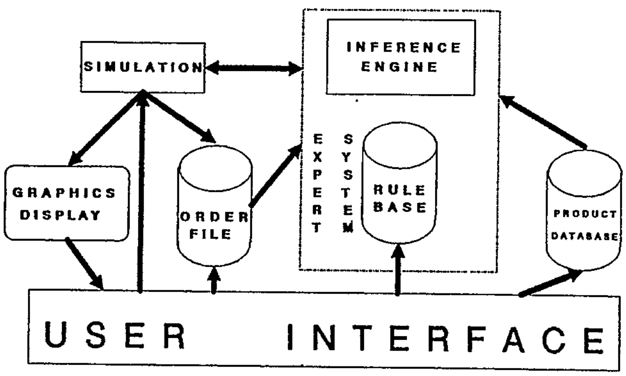

FIGURE 9. ES/Simulation System Overview

4. CONCLUSIONS

The present paper has discussed some of the many issues underlying the development of ISSEs and has provided an overview of the CASM environment.

Tobin et al. (1980) were referring to the OR scientist when they wrote 'When he constructs a tool for the manager and this is rejected, the tool stays on the shelf while the client repeatedly does without it, reminding the OR scientist that something is wrong and inviting him to look at the tool in its context, wondering if improvements are still possible'. The observation could equally be directed at the simulation modeller whose model lies untouched or recommendations rest unimplemented or, indeed, the ISSE developer whose designs fail to move the modeller from his line-editor and FORTRAN compiler.

As any would-be ISSE developer should, the CASM group take that 'shelf' seriously. The fondly-crafted tools which find their way there and those which do not must be examined and lessons learnt. Ideas, enthusiasm and theory are not enough. The proving of ISSE developments as a whole lies in their eventual impact on the practice of modelling and simulation.

REFERENCES

Ackoff,R.L.(1979),"The Future of Operational Research is Past," Journal of the Operational Research Society Vol 30 2 93-104.

Au,G.(1987), Simulation Specifications using Graphics on the Apple Mackintosh, M.Sc.(OR) project report.

Balci,O.(1986), "Requirements for Model Development Environments," Computers and Operations Research, 13 1 Jan-Feb 53-67.

Balmer,D.W. and R.J.Paul(1985), "The Direct Fire Battle: A Structured Simulation Model in Pascal", CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Balmer,D.W. and R.J.Paul(1986), "CASM - The Right Environment for Simulation," Journal of the Operational Research Society 37, 443-452.

Balmer,D.W.(1987a), "Small Scale Code Generation in a Customised Editor," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Balmer,D.W.(1987b), "Software Support for Hierarchical Modelling," Presented at the ESC Conference in Methodology and Validation, Florida

Balmer,D.W.(1987c), "Hierarchical Modelling in Discrete Event Simulation," to appear.

Balmer,D.W. and Mireskandari(1987) "Notes on a Software System to Support Hierarchical Modelling," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Bronowski,J.(1973),The Ascent of Man,British Broadcasting Corporation.

Chew,S.T., R.J.Paul and D.W.Balmer(1985), "Three Phase Simulation Modelling using an Interactive Simulation Program Generator," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Christy,D.P. and H.J.Watson(1983), "The Application of Simulation: A Survey of Industry Practice," Interfaces, 13 5 Oct 47-52.

Clementson,A.T.(1982), Extended Control and Simulation Language, Cle.Com. Ltd., Birmingham.

Crookes,J.G.(1982), "Simulation in 1981," European Journal of Operational Research 9, 1, 1-7.

Crookes,J.G.,D.W.Balmer,S.T.Chew and R.J.Paul(1986), "A Three Phase Simulation written in Pascal," Journal of the Operational Research Society, 37.6,603-618.

Doukidis,G.I. and R.J.Paul(1986), "Experiences in Automating the Formulation of

Discrete Event Simulation Models," In AI Applied to Simulation, E. J. H. Kerckhoffs, G. C. Vansteenkiste and B. P. Zeigler Eds. Simulation Councils, Inc., San Diego, Calif., pp79-90.

Doukidis,G.I. and R.J.Paul(1987), "SIPDES : a simulation program debugger using an expert system," CASM Research Report , Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

El Sheikh,A.A.R., R.J.Paul, A.S.Harding and D.W.Balmer(1987), "A Microcomputer Based Simulation Study of The Number of Berths Required At A Port",Journal of the Operational Research Society, 37 .

Goodman,D.H., D.W. Balmer and G.I. Doukidis (1987), "Interfacing Expert Systems and Simulation for Job-shop Production Scheduling", in Proceedings of the Third Expert Systems Conference, London

Goodman, D.H. and D.W. Balmer(1987), "User Interfacing for Discrete Event Simulation," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Henriksen, J. O. (1983), "The Integrated Simulation Environment (Simulation Software of the 1990's)," Operations Research 31, 6 (Nov.-Dec.), 1053-1073.

Lehman, A., B. Knodler, E. Kwee and H. Szczerbicka (1986) "Dialog-oriented and Knowledge-based Modelling in a Typical P. C. Environment," In AI applied to Simulation, E. J. H. Kerckhoffs, G. C. Vansteenkiste and B. P. Zeigler Eds. Simulation Councils, Inc., San Diego, Calif., 91-96.

Mathewson,S.C. (1980),"User Acceptance: Design Considerations for a Program Generator," Research Report, Imperial College, London.

Nance, R.E. (1979), "Model Representation in Discrete Event Simulation: Prospects for Developing Documentation Standards," In Current Issues in Computer Simulation R. Adam and A. Dogramaci (Eds) Academic Press, New York.

Nance, R.E. (1983), "A Tutorial View of Simulation Model Development" in Proceedings of the 1983 Winter Simulation Conference 325-331.

Nance, R.E. (1987), "The Conical Methodology: A Framework for Simulation Model Development," in Methodology and Validation, O. Balci (Ed.) Simulation Series Vol.19,No 1

O'Keefe, R.M. (1986), "The Three-phase Approach: A comment on strategy related characteristics of discrete event languages and models", Simulation 47:5, 208-211.

Paul,R.J.(1986),"Graphical Simulation Model Formulation", CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Paul,R.J.and S.T.Chew(1987),"Simulation Model Specification," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Pidd,M and R.N.Woolley (1980),"A Pilot Study of Problem Structuring," Journal of the Operational Research Society, Vol 31 12 1063-1068.

Reese, R. and S. Sheppard (1983), "A Software Development Environment for Simulation Programming," In Proceedings of the 1983 Winter Simulation Conference 418-426.

Rozenblit,J.W. and B.P.Zeigler (1985), "Concepts for Knowledge Based System Design Environments," in Proceedings of the 1985 Winter Simulation Conference, San Francisco

Schruben, L. (1983), "Modelling Systems using Discrete Event Simulation," in Proceedings of the 1983 Winter Simulation Conference, 101-107.

Smart,P.J. and N.J.C. Baker (1984), "SYSMOD - An Environment for Modular Simulation," In Proceedings of the 1984 Summer Simulation Conference 77-82.

Soloman,S.L. (1980),"Building Modelers: Teaching the Art of Simulation," Interfaces, 10 2 65-72.

Standridge, C.R. and S. A. Walker (1983), "An Introduction to Simulation Support Software," In Proceedings of the 1983 Winter Simulation Conference 381-384.

Tobin,N.R., Rapley,K. and W.Teather (1980),"The Changing Role of OR,"Journal of the Operational Research Society, Vol 31 4 279-288.

Zeigler, B. P.(1984), Multifacetted Modelling and Discrete Event Simulation, Academic Press, London.

Zeigler, B. P. and L. de Wael (1986), "Towards a Knowledge-based Implementation of Multifacetted Modelling Methodology," In AI applied to Simulation, E. J. H. Kerckhoffs, G. C. Vansteenkiste and B. P. Zeigler Eds. Simulation Councils, Inc., San Diego, Calif. 42-51.