Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

INTEGRATED SUPPORT ENVIRONMENTS FOR SIMULATION MODELLING

David W. Balmer
Ray J. Paul

Department of Statistical and Mathematical Sciences
London School of Economics and Political Science
Houghton Street, London, WC2A 2AE, United Kingdom

ABSTRACT

The general concept of an integrated software support environment to enhance the productivity of the simulation modeller has motivated several groups to design and implement such environments. Underlying each such development is a perspective, implicit or explicit, of the methodology of simulation modelling.

This paper reviews the experience of the Computer Aided Simulation Modelling (CASM) group at the London School of Economics in investigating integrated modelling support and discusses the methodological perspectives underlying the work.

1. INTRODUCTION

The keynote address of the Winter Simulation Conference is a place where the speaker usually reflects on the past successes of simulation modelling and, looking to the future, identifies the critical challenges facing researchers. In 1987 the keynote speaker, White, commenting on the critical importance of simulation modelling in the design and operation of complex, integrated systems (for materials handling), suggested the way forward would involve making the approach more accessible to managers and engineers [White 1987].

In 1989 Pritsker identified simulation modelling as a pragmatic approach to problems with an outstanding track record, [Pritsker 1989]. He attributed the successes to the skills and attitudes of the modellers, a methodology which combines 'the engineering concepts of design and control with the experimental approach of the scientist' and the range of 'more and better tools to do the job'.

The general concept of an integrated software support environment (ISSE) for discrete event simulation has been advanced by many authors [Nance 1979; Henrikson 1983; Standridge and Walker 1983; Rozenblit and Zeigler 1985; Balci 1986; Balmer and Paul 1986; Nance 1987] as a way of providing 'yet more and better tools'. Prototypes of such environments have been constructed with common intention to enhance the productivity of simulation modellers. By formalising the processes of simulation modelling, creative and technically routine, and fixing these in software, such environments have the potential to improve the productivity of the specialist modeller but also to effect some transference of modelling skills and methodology to a wider class of users as envisaged by White.

Of course, simulation modelling takes place in a variety of contexts and serves a wide range of purposes. Although the general modelling perspectives offered by the simulation approach are broadly applicable and flexible, they find realisation in many alternative formalisms. Models may be implemented in different general purpose and specialist languages each with its own attractions, range of applicability and proponents.

These diversities are reflected in the development of ISSEs. There are areas of broad agreement concerning the general functional components of an ISSE but important differences of view remain in regard of detail, structure and directions of development.

In some cases design is guided by an explicit and detailed methodological perspective [Nance 1987; Balci and Nance 1987a; Zeigler 1976,1984; Rozenblit and Zeigler 1986]. In other cases the necessity to support emerging technologies such as parallelism may determine developments [Unger et al. 1986]. Balmer attempts a brief catalogue of the factors of development context and modelling perspective which may influence the design and construction of an ISSE [Balmer 1987d].

The present paper describes developments towards a loosely integrated software support environment within the CASM group at the London School of Economics and relates these to a more general discussion of modelling and simulation support environments.

The next two sections pursue the general discussion of modelling perspective and software support. These are followed by sections on developments within the CASM project. A penultimate section reflects on the issues raised and the final section draws some brief conclusions.

2. MODELLING PERSPECTIVES

2.1 The Modelling Process

The descriptions of the simulation modelling process provided in standard textbooks or in tutorial papers usually involve a series of steps with some possibility of cycling.

Schruben [1983] reviews a number of such descriptions and provides a tabular summary. There are common emphases on 'formulate the problem', 'build the model' and 'analyze the model'. These crude accounts are not uninformative but in terms of providing a pattern for action or preliminary definition of requirements for an ISSE, they leave many questions unanswered and much room for alternative interpretation.

2.2 Modelling Methodologies

Mathewson[1989a] discusses the simulation modelling process under the three broad areas of problem definition, model development and decision support identified in Figure 1, listing desiderata and the ways in which simulation languages have attempted to provide for these. As he comments not only does the figure draw attention to these broad areas and indicate within each a sequence of component issues but the high level of interconnection stresses the complexity and interaction of modelling activities.

Nance [1987] adds to this a methodology for model development, the 'conical methodology', of top-down model definition and bottom-up model specification.

An alternative formalism of the modelling and simulation process is derived from a system-theoretic base by Zeigler [1976,1984] and a methodology for model development - 'a top-down design resulting in a hierarchically constructed model is the basic methodology compatible with the multifacetted-system approach'.

2.3 Diversities of Modelling Purpose and Context

2.3.1 Diversity of Product

The product of the modelling and simulation process should be insight for action rather than simply the software system that is the model. Ackoff[1979] stresses the importance of the

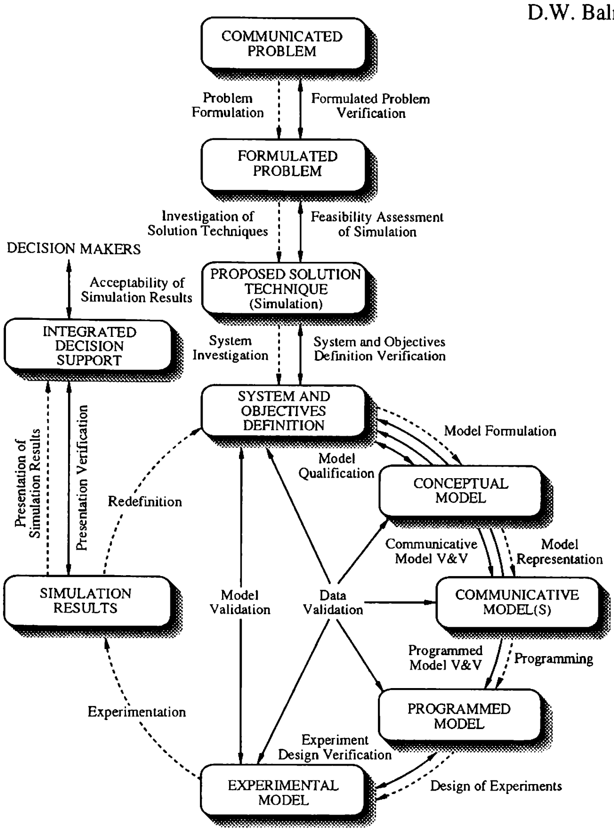

Figure 1. The Life Cycle of a Simulation Study (from [Balci 1986])

process of modelling and the insights so generated as the principal product. The role of the modeller and the prime requirement of an ISSE in this context is thus one of facilitator enabling the more adequate participation of "stake holders" in the modelling process. The social dimension of the process could be more important than the quality of code produced or the resolution of any statistical experimentation, per se.

2.3.2 Diversity of Systems Modelled

The components of the system modelled may be mainly mechanical, electrical or electronic and their behaviour, although complex, stable and subject to known physical laws. Other systems may involve mainly 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 Operational Research. Problem solutions, model-based or otherwise, have a short life [Ackoff 1979; Tobin et al. 1980]. The pace of model development and use have to match that of the decision-maker's environment.

The anticipation of a rapidly changing 'messy' modelling context leads to an emphasis on rapid, interactive model development and investigation with a concern for the support of the 'process' as well as the 'model' concept of 'modelling product'.

2.4 Practical Application Contexts

The specific challenge issued by Pritsker, [Pritsker 1989], was that case studies be written up so that the practical problems encountered could form a background for academic research. The area of simulation modelling is solidly practical and although it is only appropriate that the academic researcher should speculate on the abstract and theorise about the activity it is important that such speculation should be grounded in the practical context of real modelling. Case studies where written up, are rarely presented in forms which reveal the patterns of thought underlying the modelling process nor the full range of difficulties encountered.

2.4.1 CASM Application Context

The CASM group has endeavoured to keep at least one foot on the ground of practical experience by personal involvement in a range of problems. These include military modelling - armour vs armour battle study [Balmer and Paul 1985], submarine interception simulation/gaming study [Balmer et al. 1981; Douglas and Balmer 1983], port operations classical and container port [El Sheik et al 1987; Balmer and Goodman 1988), jobshop [Goodman et al. 1987], police manpower management [Balmer et al. 1989], marketing - design and evaluation of advertising campaigns [Balmer 1990a], hospital outpatients clinics [Kuljis et al. 1990] and transport and traffic research [Tosic et al. 1990].

3. SUPPORT ENVIRONMENTS

3.1 Commercial Support Environments

A recent analysis of the variety of computing environments designed to support simulation modelling is provided by Mathewson [1989b]. His focus is primarily on those systems which have achieved a commercial realisation although he includes reference to current academic research activity in the area.

He classifies systems as

generalised graphical building blocks
with transaction flow
application-specific systems
simulation preprocessors
and support environments.

The first category is exemplified by GPSS, SLAM and SIMAN languages. These systems allow 'programming' in terms of a stylised description of a flow diagram, thus avoiding the need to write a full computer program. The discussion of application-specific systems includes PCModel, SimFactory, MAST, SAME, Witness, XCELL and ModelMaster. Many of these systems are data-driven generic models with specialised text and graphics editors to build and edit the data files describing the model and its displays.

The two principal examples of simulation preprocessors or program generators are Draft and CAPS/ECSL. These are both based on the specification of a model in terms of an entity or activity cycle diagram and translation into error-free source code in FORTRAN or ECSL for compilation and execution or subsequent development.

The commercial simulation support environments

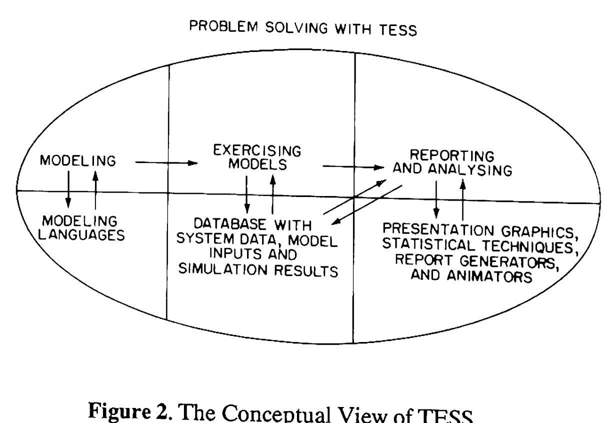

Figure 2. The Conceptual View of TESS (from [Standridge 1986])

considered include TESS, SDL, RESQ and RESQME. Figure 2 provides a conceptual view of problem solving with TESS. The modeller builds a model using the underlying modelling languages - SLAM II, MAP/1 and GPSS/H and designs and executes model experiments using the TESS BUILD commands and performs statistical analyses, reports results and summaries [Standridge 1986].

3.2 Research Support Environments

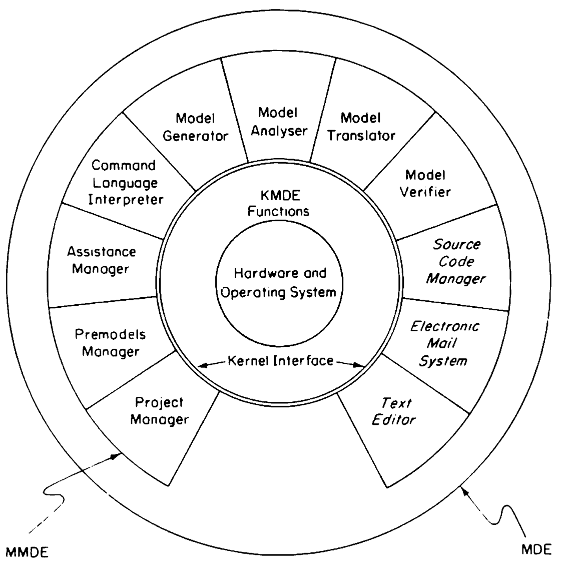

Figure 3. The Structure of Model Development Environments (from [Nance 1987])

Figure 3 indicates the basic building blocks for an ISSE and in particular for the Simulation Modelling Development Environment (SMDE)[Balci and Nance 1987a]. This design intends a more comprehensive support of all aspects of the simulation modelling process and builds on the conical methodology of [Nance 1987].

Rozenblit and Zeigler [1987] define an ISSE built around the DEVS formalism [Zeigler 1984]. This emphasises the synthesis of models based on the system entity structure encompassing all boundaries and decompositions previously conceived in respect of the system to be modelled. Modelling objectives determine generic frames which are used to prune the system entity structure to serve as skeletons for the synthesis of models fulfilling current modelling objectives.

4. BASIC CASM SUPPORT ENVIRONMENTS

4.1 Basic Design Considerations and Development Strategies

The development of an ISSE is itself a not-insubstantial software project which demands the definition of requirements, specifications and design.

4.1.1 Specific Hardware and Operating Contexts

Some consideration of hardware and operating system for development and application is inevitable and design considerations must include notions of portability.

The considerations which have led to the larger part of the CASM development being under taken on the IBM PC range from the original PC, PC AT and more recently on PS/2 60s and 70s are principally practical.

Such machines have been readily accessible to both academics and research students involved in the project and are available to other students, undergraduate and postgraduate, who have 'tested' and 'evaluated' software developments in their course work and in associated projects.

The IBM PC has also set the standard for hardware which is readily available to the 'targeted' users to be found in the OR practitioner groups in public and private corporations, in government, local and national and in public bodies such as health authorities. Research sets its own agenda and should be forward looking, anticipating future hardware development. Nevertheless, if software support facilities are to have a practical impact on the activities of such a user group, they do well not to demand esoteric hardware or non-standard operating systems.

However, the PC environment has offered relatively slow processing speeds, limited memory, direct and indirect, and modest graphical capabilities. DOS is essentially a single-user operating system without facilities for multitasking. The windowing environments which have become commonplace on most workstations have taken some time to emerge on the PC. Exploratory work using the GEM and MS Windows environments has only touched on the potential of this framework.

These factors have made it necessary to craft tools to achieve interfaces with proprietary software or to mimic multitasking which have been better drawn from the toolbox of a more satisfactory operating environment.

Most of the development work has focused on structured general purpose languages, Pascal, Modula2 and C and simulation systems developed in-house in these languages.

4.1.2 Design and Development Strategy

The modelling perspectives shared by the group emphasise the operational research context, with 'messy' problems occurring in people-oriented environments which are rarely stable in the long term. The concern is with the impact of the 'process' of modelling as well as the 'model' produced.

The group has not the confidence in a single modelling methodology except in the broadest of terms to guide the design and development on an ISSE but regards the development of ISSEs as a means to explore methodological issues. Methodological insights derive from varied individual experiences in practical application contexts.

The design and development strategy adopted is based extensive exploratory prototyping of tools in a loosely integrated environment with only a limited regard for long-term compatibility or portability.

4.2 Basic System Components

Figure 4 shows the basic system components of a minimal ISSE for simulation. Its initial emphasis is on moving towards the implementation of the automation paradigm [Balci and Nance 1987b]

4.2.1 Basic Simulation Environment

Model execution facilities are centred around a basic simulation system consisting of libraries of declarations, functions and procedures and a program template in suitable high-level general purpose language; Pascal systems have tended to dominate but similar approaches have been pursued in Modula2 and C. These provide a structured framework for the development of executable simulation models with other facilities such as file handling, text editor, compiler, linker, debugger accepted from the host environment. Clearly all native support facilities such as editors, debuggers and help systems are geared to the base language and offer little or no support in respect of the specialist library functions. This deficiency is remedied by additional support tools some of which are described below. A version used extensively for teaching is implemented in TURBO Pascal but the basic system exists in a variety of mini and microcomputing environments. It offers the advantages of structure and modularity with the flexibility and support of a widely used high level language.

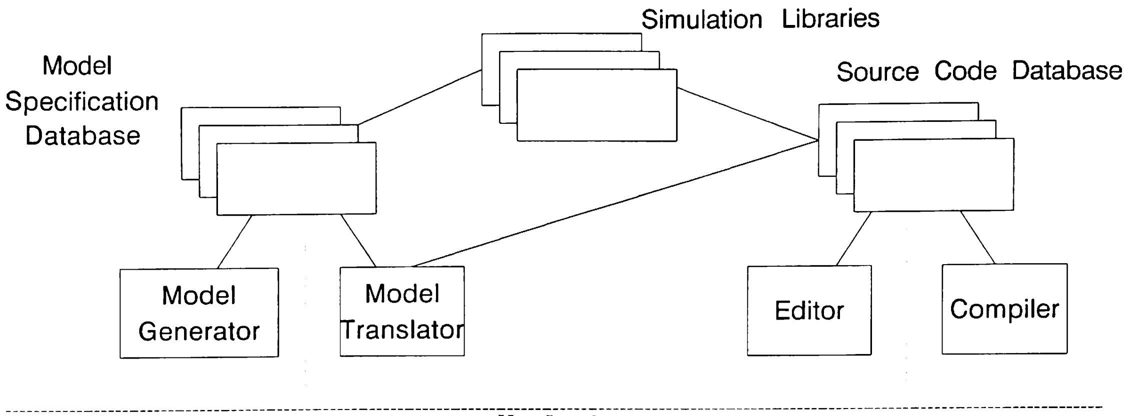

Figure 4. The Structure of a Minimal Support Environment

The principal 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; OKeefe 1986]. The three phase method seems to offer clear advantages in the areas of modelling clarity, model maintainability and modularity.

The basic functionality and structure of the system is subject to continuing development: some versions include animation facilities or extended statistical functions. A substantial account of an early version of this basic software environment can be found in [Crookes et al. 1986].

4.2.2 Automatic Programming

The automation paradigm shifts the principal focus from the executable formalism back to a higher level model specification. The paradigm requires the generation and manipulation of model specifications and the translation of model specifications into executable form. The form of simulation preprocessing has found implementation under the heading of interactive simulation program generators,(ISPGs). These combine the entry of the formal specification of a simulation model with its translation into appropriate source code of an executable model or direct interpretation of the specification. The CASM project has produced a number of ISPGs.

For instance, Chew et al. [1985] give details of the prototype ISPG, LANGEN, which, in external form and concept, owed much to earlier pioneering work of Clementson [1982]. 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 simulation system described above. The description of the ACD is communicated to each through a similar interrogative dialogue.

4.2.3 Model Translation

The translation of a formal model specification into executable source code employing the data structures and procedures and functions of the basic simulation environment has, as indicated above, been implemented within a sequence of prototype ISPGs. The structure of translation within these prototypes has drawn less than it might on the formal methods of compilation. The translation problem has rarely presented difficulties and the evolution of the prototypes has been conditioned more by changes in the target language and its structures and by developments in the formal model specification language.

In addition to the translation of the model specification into the executable source code, the direct execution of the specification through an interpreter has been explored.

Further translation of the model specification file into code capable of analyzing the model as a Markov chain offers for models of limited size an alternative analytical/numerical evaluation which can be set alongside any simulation model produced.

4.2.4 Model Specification

Again, the production of a model specification file has been often been supported as an integral part of an ISPG. Different prototypes have employed a variety of enquiry mechanisms from the direct interrogation of the user referred to above to form filling [Knox 1988; Balmer 1988], natural language understanding systems [Doukidis 1985; Barakat and Paul 1989] and graphics-driven [Au 1990; Au and Paul 1990]. Although translation of specification has usually been into a three phase world view, the model specification language of the user presumed in these has emphasised a variety different forms eg entity life cycles, activity structures, event structures, corresponding loosely to the principal world views of simulation modelling. The model specification files produced have also taken a number of different forms some of which have been readily amenable to direct editing within standard text editors.

Diagramming techniques such as the ACD offer a simple representation of basic simulation model structure which is widely accessible and easily communicable. More formal or mathematical approaches have virtues of completeness and precise but tend to be more cumbersome and less accessible. The exploration of simulation model specification methods combining both the system-theoretic formalisms of Zeigler [1984], formal languages Z and VDM and diagrammatic techniques of ACDs or Petrinets is described in [Ceric and Paul 1989; Paul and Ceric 1990; Domingo and Paul 1990].

4.3 Further Specifics

4.3.1 Support for Editing

The text editor provided by a host environment is unlikely to have any features of specific its role in model development. Within the TURBO Pascal environment, extra functionality may be provided through a utility which is accessible through single alt keystrokes [Balmer 1987a]. The utility makes available specific on-line help tailored to the simulation system and localised automatic programming through a 'form filling' generation of basic program structures such as B and C Procedures for insertion in the program under development within the TURBO editor.

4.3.2 Support for Debugging

TURBO Pascal 5.0 includes an interactive debugger which offers powerful general support to the debugging of program code. For the modeller, however, this offers no direct support for the coding errors specific to simulation modelling using the basic simulation systems. Doukidis and Paul [1986] describe 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.

Much of the information required by SIPDES of the user can be obtained by direct inspection of the source code of the model. A further support tool recently developed acts directly on the source code, analyzing its structure and returning the results of this analysis [Balmer 1990]. The tool is able to identify potential errors, omissions and redundancies and subject to user confirmation to act directly to rectify these. It is not as yet integrated with SIPDES but such an integration would offer a powerful, modelling-specific, debugging system.

4.3.3 Disassembling a Simulation Program

At current levels of realisation it is estimated that the automation paradigm results in the automatic programming of only 80% of the model [Nance and Balci 1987]. The analysis of the Pascal simulation program which is produced by debugging aid reported above produces for its internal management of the debugging process requires but a minor addition to produce as output a model specification file to complete the translation circle at least in respect of those model components that can be readily expressed in the formalism used. Further, though minor, modification would allow additional Pascal fragments to be added to the model specification form to retain the entire substance of a generated model which had been modified by the addition of further Pascal code.

4.3.4 Semantic Nets For Model Specification

The system described by Barakat and Paul [1989] adopts a spreadsheet interface approach. It allows a description of the simulation problem to be written in natural language form and the system will interpret this description using semantic networks. One level of the spreadsheet is the natural language description, a second is the semantic network inferred from the description and a third level provides an activity cycle diagram for the problem. Further levels may contain the numerical data required to actually run the model.

4.3.5 Output Analysis

The investigations of the comparative performance of statistical procedures such as confidence intervals based on non- and overlapping batch means, spectral methods and standardised time series methods has produced tools for their application to simulation output [Kevork 1990]. Other work has investigated the interfacing with standard statistical packages such as Minitab and GLIM with special macros to implement non-standard simulation analyses.

5. FURTHER DEVELOPMENTS

5.1 Graphics-Driven Systems

The latest development in the series of ISPG prototypes, MacGraSE, is described by [Au 1990; Au and Paul 1990]. For this system development has moved onto the Apple Mackintosh. This graphics-driven environment allows users, the analyst and the client, to specify the model using iconic representations for system objects. The icons are assembled on the screen in a descriptive fashion, intelligible to client and analyst. The layout does not follow any particular formalism or diagramming convention, although underpinning the method is the activity cycle diagram concept. The system provides the user with assistance in the definition of the basic model logic and the

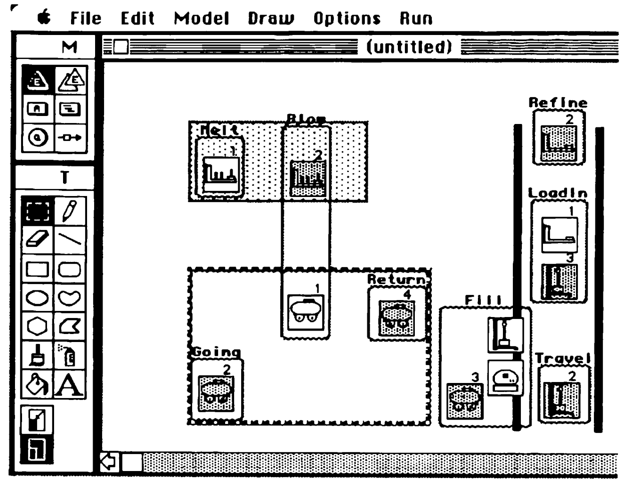

Figure 5. An Example of a Model Construction Picture for MacGraSE

addition of quantitative and supplementary conditional information. Figure 5 shows a model under construction. Entities specified in the system are associated with an easily identifiable icon. A description of the characteristics of the entity may be input to a table. The system allows the user to define the model logic in terms of entity life cycles and to draw the associated activity cycle diagram. The system allows the model to be executed in interpretive mode so that the dynamics of the system can be checked and verified in terms of the visual display defined. MacGraSE will generate a Pascal source code file for the model which can be developed further to accommodate any specific features whose complete specification in terms of the current graphics-driven environment had proved impossible.

5.2 Interfacing Simulation and Expert Systems

Many modelling contexts imply a need for flexible representation of complex management control options. The standard simulation modelling structures have often been observed to be inadequate for this purpose. It is valuable to separate the more formal and stable aspects of the model from the aspects of management control with which the decision-maker wishes to experiment. The former may be well represented in the standard simulation modelling structures; the latter in the form of an expert system or systems and the two components suitably interfaced.

The three phase world view fits in with such decomposition particularly well.

The C phase involves the testing of the current model state against a set of conditions or rules which will determine which, if any, new activity starts are implied. If the conditions are complex or the rule systems interrelated and evolving during the simulation study the condition check might well be effected within a rule-based expert system. The simulation executive consults an 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.

Developments of this type within the CASM environment have been explored 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 and of a container port where the exploration of alternative stack management rules is the primary modelling objective [Goodman et al. 1987; Balmer et al. 1989].

6. REFLECTIONS

6.1 Alternative Paradigms

Most textbooks are very weak in their exposition of how to formulate problems. Where a more extensive methodology has been expounded the methods seem to be highly structured.

How do highly structured methods fare in 'messy' practical problems?

The paradigms underlying these methods seem to be largely those of software engineering. Indeed, some views of the modelling process do not differentiate between simulation model development and general software development [McKay et al. 1986]!.

Pidd and Wooley [1980] observed, in a slightly different context, that the highly formalised approach to problem structuring '... was not observed in the O.R. groups we visited.'

Are structured modelling methodologies used in practice?

Are there other paradigms to consider which are less rigidly structured and emphasising exploratory approaches?

One can find wisdom in unusual places! There is much that can be applied to modelling in the 'Way of Strategy':

'Whenever we have become preoccupied with small details, we must suddenly change into a large spirit, interchanging large with small. This is one of the essences of strategy.'[Mushashi 1645 p81]

'In strategy it is important to see distant things as if they were close and to take a distanced view of close things.' [Mushashi 1645 p54].

6.2 Testing Methodologies

How does one test modelling wisdom?

Henz-Luehrmann and Byrkett [1989] report the results of a controlled experiment to examine the benefits of animation over tabular/graph output in a simulation study to support decision-making. They were unable to display any tangible benefits.

The literature of simulation modelling does not abound with reports of empirical studies on the relative effectiveness of particular software support tools and of the methodological perspective they embody. This is scarcely surprising. The design and execution of adequate testing experiments would be hugely expensive and any conclusions would be limited in their scope and validity. Perhaps the answer is provided by paraphrasing Tobin et al. [1980] - 'When the ISSE developer constructs a tool for the modeller and this is rejected, the tool stays on the shelf while the modeller repeatedly does without it, reminding the developer that something is wrong and inviting him/her to look at the tool in its context, wondering if improvements are still possible'.

7. CONCLUSIONS

The present paper has discussed some of the many issues underlying the development of ISSEs and has illustrated these in an account of developments within the CASM project.

REFERENCES

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

Au, G.(1990), "A Graphics-Driven Approach to Discrete Event Simulation," Ph.D. Thesis, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Au, G. and R.J. Paul (1990), "A Complete Graphical Discrete Event Simulation Environment," Presented at the *Young OR Conference*, Warwick, UK.

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

Balci, O. and R.E. Nance (1987a), "Simulation Model Development Environment: A Research Prototype," *Journal of the Operational Research Society* 38, 8, 753-763.

Balci, O. and R.E. Nance (1987b), " Simulation Support: Prototyping the Automation Paradigm," In *Proceedings of the 1987 Winter Simulation Conference*, A. Thesen, H. Grant and W.D. Kelton, Eds. IEEE, Piscataway, NJ, 495-502.

Balmer, D.W., J.G. Crookes and A.M.M. Douglas (1981), "SUES and MicroSUES - A Study of Submarine Interception Tactics," Report to Ministry of Defence, UK.

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, 5,* 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," In *Proceedings of the 1987 UKSC Conference on Computer Simulation*, R.N. Zobel, Ed. UKSC, 13-18.

Balmer, D.W. (1987d), "Modelling Styles and Their Support within the CASM Environment," In *Proceedings of the 1987 Winter Simulation Conference,* A. Thesen, H. Grant and W.D. Kelton, Eds. IEEE, Piscataway, NJ, 478-485.

Balmer, D.W. (1988a), "Automating the Analysis of a Markov Chain Model based on a Simulation Model Specification," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Balmer, D.W. (1988b), "A Simple Program Generator: User Manual," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Balmer, D.W. and D.H. Goodman (1988), "The Simulation Study of a Container Port," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, UK.

Balmer, D.W. (1990a), "AdTest - A Simulation Model to Evaluate the Sales Effects of Advertising," CASM Research Report, Department of Statistical and Mathematical Sciences, London School of Economics, UK.

Barakat, M. and R.J. Paul (1989), "Simulation Modelling using Semantic Nets," Presented at the *Annual OR Conference,* University of Southampton, UK.

Ceric, V. and R.J. Paul (1989), "Preliminary Investigations into Simulation Model Representation," In *Proceedings of the 11th Symposium on Computer at the University*, Cavtat, Yugoslavia.

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.

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.

Crookes, J. G. (1987) 'Generators, Generic Models and Methodology.' *Journal of the Operational Research Society*, 38, 8, 765-768.

Domingo, L.T. and R.J. Paul (1990), " A Introduction to Simulation Specification using Formal Methods," Presented at the *Young OR Conference,* Warwick, UK.

Douglas, A.M.M. and D.W. Balmer (1983), "Microcomputer Submarine Encounter Simulation," Report to the Ministry of Defence, UK.

Doukidis, G.I. (1986), "Discrete Event Simulation Formulation using Natural Language Understanding Systems," Ph.D. Thesis, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

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* 38, 8, 673-681.

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.

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

Henz-Luehrmann, M.A. and D.L.Byrkett (1989) 'A Pilot Study of the Impact of Animation on Decision-making.' *Simulation* Vol 53 no4 Oct p153-158.

Kevork, I. (1990), "An Investigation of Confidence Interval Methods for the Analysis of Discrete Event Simulation Output," Ph.D. Thesis, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Knox, P.M. (1988), "Graphics Based Automated Simulation Modelling," Ph.D. Thesis, Department of Statistical and Mathematical Sciences, London School of Economics, London, England.

Kuljis, J., R.J. Paul, H. Malin and S. Thakar (1990), "Designing an Out-Patient Clinic Modelling Package," In *Proceedings of the 12th Symposium on Computer at the University*, Cavtat, Yugoslavia.

McKay,K.N. J.A.Buzacott, J.B Moore and C.J.Strong 1986 Software Engineering applied to Discrete Event Simulations in *Proceedings of the 1986 Winter Simulation Conference* J. Wilson,J.Henriksen and S. Roberts (eds)pp485..493.

Mathewson, S.C. (1989a), "The Implementation of Simulation Languages," In *Computer Modelling for Discrete Event Simulation,* M.Pidd, Ed. J. Wiley and Sons, London, 23-56.

Mathewson, S.C. (1989b), "Simulation Support Environments," In *Computer Modelling for Discrete Event Simulation,* M.Pidd, Ed. J. Wiley and Sons, London, 57-100.

Mushashi, M. (1645) *A Book of Five Rings,* English Translation: V. Harris, Allison and Busby Limited, London 1982.

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*, J. Banks and B. Schmeiser, Eds. IEEE, Piscataway, NJ, 325-331.

Nance, R.E. (1987), "The Conical Methodology: A Framework for Simulation Model Development," In *Methodology and Validation,* O. Balci Ed. Simulation Series 19, 1, 38-43.

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.

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

Paul, R.J. and V. Ceric (1990), "Simulation Model Representation Methods," *to appear*.

Paul, R.J. and S.T. Chew (1987), " Simulation Modelling using an Interactive Simulation Program Generator," *Journal of the Operational Research Society* 38, 8 735-752.

Pritsker, A.A.B. (1989), "Why Simulation Works," In *Proceedings of the 1989 Winter Simulation Conference,* E.A. MacNair, K.J. Musselman and P. Heidelberger, Eds. IEEE, Piscataway, NJ, 1-6.

Rozenblit, J.W. and B.P. Zeigler (1985), "Concepts for Knowledge Based System Design Environments," In *Proceedings of the 1985 Winter Simulation Conference,* G.C. Blais and S. Solomon, Eds. IEEE, Piscataway, NJ, 223-231.

Rozenblit, J.W. and B.P. Zeigler (1986) "Entity-based Structures for Model and Experimental Frame Construction," In *Modelling and Simulation in Artificial Intelligence Era*, M.S. Elzas et al. Eds. North Holland, Amsterdam.

Rozenblit, J.W. and B.P. Zeigler (1987) "Multifaceted Methodology: A Framework for Knowledge-based Simulation Modelling," In *Methodology and Validation,* O. Balci Ed. Simulation Series 19, 1, 50-53.

Schruben, L. (1983), "Modelling Systems using Discrete Event Simulation," in *Proceedings of the 1983 Winter Simulation Conference,* J. Banks and B. Schmeiser, Eds. IEEE, Piscataway, NJ, 101- 107.

Standridge, C.R. and S.A. Walker (1983), "An Introduction to Simulation Support Software," In *Proceedings of the 1983 Winter Simulation Conference*, J. Banks and B. Schmeiser, Eds. Piscataway, NJ, 381-384.

Standridge, C.R. (1986), "A Tutorial on TESS: The Extended Simulation System," In *Proceedings of the 1986 Winter Simulation Conference*, J. Hendriksen and S.D. Roberts, Eds. Piscataway, NJ.

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

Tosic, V., R.J. Paul and V. Ceric (1990), "Simulation as a Decision Aiding Tool in Transport Engineering", Presented at the *IFORS Xii*, Athens.

Unger, B., J. Cleary, G. Lomow, K. Sind and Li Xining (1986) "Jade Virtual Time-Implementation Manual," Research Report #86/242/16, Computer Science,University of Calgary.

White, J.A. (1987), "Simulation: Pushing a Dead Mouse Through a Maze?" In *Proceedings of the 1987 Winter Simulation Conference*, A. Thesen, H. Grant and W.D. Kelton, Eds. IEEE, Piscataway, NJ, 1-3.

Zeigler, B.P. (1976), *Theory of Modelling and Simulation*, J. Wiley and Sons, New York.

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,