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

WHAT IS VISUAL INTERACTIVE SIMULATION ?

(AND IS THERE A METHODOLOGY FOR DOING IT RIGHT ?)

Robert M. O'Keefe

Department of Decision Sciences
and Engineering Systems
Rensselaer Polytechnic Institute
Troy, New York 12180

ABSTRACT

Visual Interactive Simulation (VIS) is the development and application of simulations which produce a dynamic display of the system model, and allow the user to interact with the running simulation. This paper presents the basic ideas behinds VIS, and the methods whereby it can be achieved. The benefits of using VIS, and folklore about the advantages and problems of VIS, are presented. It is concluded that amongst those practicing VIS a number of competing decision aiding methodologies exist, and that a single unifying methodology is necessary for the advancement of VIS.

1. INTRODUCTION

A Visual Interactive Simulation (VIS) is a simulation which produces a dynamic display of the system model, and allows the user to interact with the running simulation. Although very early examples of VIS can be found, for example Amiry (1965), the present approach to simulation known as VIS dates back to the seminal work of Hurrion (Hurrion and Secker, 1978).

There are a number of published reviews of VIS, of which Hollocks (1984) and Bell and O'Keefe (1987) are both good introductions. Papers which present specific software systems include Fiddy et. al. (1981) on SEE-WHY, Melamed and Morris (1985) on the Performance Analysis Workstation, and O'Keefe (1987) on Inter_SIM. An excellent discussion on the application of VIS can be found in Macintosh et. al. (1984).

Typically, a VIS system or application provides facilities for :-

(1) *Visual Output*: portraying the dynamic behavior of the system model.

(2) *User Interaction*: allowing the user to interact with the running model. Interaction can be *model determined*, where the simulation halts and requests input from the user, or *user determined*, where the user halts the simulation at will.

(3) *Visual Input*: where a model can be created visually instead of being programmed or data-driven.

At present, (1) is absolutely necessary to describe a system as VIS, (1) and (2) are necessary and sufficient. The majority of VIS systems do not provide (3). Animation tools such as CINEMA provide (1), and increasingly provide for some level of user interaction.

1.1 Visual Output

A display of the dynamic behavior of the system model is normally achieved through a 'mimic' diagram of the system, using icons to represent objects. There are three distinct approaches to facilities for visual output :-

(a) *Embedded Programming.* The simulation programmer codes visual output statements, in much the same way that statements for statistics collection might be added to a simulation. Examples of this approach include SEE-WHY, OPTIK (Insight International, 1986) and Pascal_SIM (O'Keefe and Davies, 1986b). This provides great flexibility, but the extra development effort can be time consuming, and can create software engineering problems, since the resulting simulation will be larger and more complex than the non-visual equivalent.

(b) *Automatic Display.* A standard type of display is provided, which can be used with little or no effort on the part of the developer. For example, the developer specifies the appearance and position of queues, but the system executive deals with the movement of icons. An example of this approach is Inter_SIM (O'Keefe, 1987). Automatic display is analogous to automatic statistics collection, and has the same advantages and disadvantages ie. it is very easy to use, but the developer can be constrained by lack of flexibility.

(c) *Animation.* The simulation automatically produces formatted output, which is then decoded by a animator, using a separately produced static background and rules for icon movement. Examples of this approach include TESS (Standridge, 1985) and CINEMA. Animation has the considerable advantage that the display can be run in parallel with or following simulation execution.

1.2 User Interaction

Similarly, there are three general approaches to handling both model and user determined interaction :-

(a) *Embedded Programming.* All available interactions are directly programmed. As above, this provides great flexibility at the cost of additional development effort.

(b) *Standard Interactions.* The package provides a set of standard user determined interactions, which may or may not be extended by the developer. An example of this approach is OPTIK.

These both have the grave disadvantage that any user determined interaction not perceived by the programmer (in (a)) or package designer (in (b)) can not be immediately provided on request.

(c) *Stopping Interpretation.* Where the underlying language is interpreted, the developer can rely on stopping interpretation and altering code. For example, this can be done with SMALLTALK (Goldberg and Robson, 1983) and most Lisp based systems. This has the advantage of great flexibility, but the disadvantage that even simple changes may require knowledge of the underlying language. In packages where a high level description of the system is interpreted, the user can alter the actual description. For example, this can be done with Inter_SIM and XCELL.

1.3 Visual Input

There are two distinct approaches to describing a simulation model using visual input :-

(a) *Graphical Representation.* A simulation is described using an intermediate graphical representation, such as an activity-cycle or network diagram. A good example of this is the network interface facilities in TESS. Essentially, this approach is close to the emerging methods of *visual programming* (IEEE, 1984).

(b) *Iconic Representation.* Here a simulation is described by placing and linking icons. Examples include the generic model facilities in SimKit (Faught, 1986), and the Performance Analysis Workstation (Melamed and Morris, 1985). This approach has the considerable advantage that the same representation can be used as the basis for visual output. The developer is constructing both the model and the display simultaneously.

2. THE BENEFITS OF VIS

The use of VIS, and the development of specialized packages for doing VIS, is obviously driven by the benefits of using VIS over traditional batch simulation methods. Essentially, VIS gives the developer and the client opportunities for three inter-related activities: *selling*, *gaming* and *learning*. Any or all of these can be beneficial in a particular situation.

2.1 Selling

VIS, and particularly visual output or animation, is a tremendous aid to selling the simulation method, a simulation model, or a specific solution. Clients can quickly understand model behavior, and can help validate the simulation by following the dynamic display. The display becomes a *communication medium* that provides a common base between developer and client for discussions on development, experimentation, etc. (Crookes, 1981; Hurrion and Secker, 1978). It is an excellent *presentation medium* for results and designs that have been obtained either using VIS or by more traditionally methods ie. statistical experimentation.

2.2 Gaming

Using model determined interaction, decision makers can be incorporated into the model, and decisions that are too difficult to encapsulate into a model can be referred to the user. The effect of following various policies can be assessed. Gaming is particularly appropriate for complex systems which are never allowed to reach steady-state due to a necessity for frequent management intervention.

2.3 Learning

In addition to being used as an analysis tool, a VIS can be used by a client to 'play' at managing a system. The benefit to the client is an increase in understanding system behavior, and perhaps some information that can help solve an ill-structured problem, rather than an actual 'solution' to a 'problem'.

3. SOME FOLKLORE

Although the benefits of VIS, discussed above, are frequently quoted, and the drive towards increased use of animation and interaction proceeds unabated, there is little or no hard evidence on the real advantages, the problems, and the disadvantages of using VIS. Objective appraisal is not helped by the acrimony that can exist between the 'animation' and 'statistics' camps. However, amongst practitioners there is an emerging *folklore* (for want of a better phrase), based upon previous experience. Some of the more substantial elements of this folklore are presented here.

(1) *Visual Output Greatly Increases Model Credibility.*

It is generally accepted that where clients can follow a dynamic display of the model, credibility is far greater. A client need no longer accept the developer's word that the model is in some sense 'valid'. In many organizations, clients will no longer accept the use of a simulation that does not provide visual output.

(2) *The Power Lies in the Interaction.*

Many clients are skeptical of a slick presentation using animation. When they can interact with the model, and see that on alteration it behaves as the real system would, credibility rockets. Model determined interaction allows the integration of client decisions with the model.

(3) *The Use of Visual Output Depends upon the Decision Situation.*

In some instances, the client is interested in following the system's behavior under *average performance.* Some studies have used constant activity durations in an attempt to portray this. However, in many studies, it is the extreme or crisis situations that are of interest. The developer must arrange for sample runs that portray these to be available for the client.

(4) *There are Not Enough Guidelines on Output Design.*

What is provided as output, and what it looks like, is frequently a function of the package being used, not the decision situation. Whilst manufacturing and service systems can be easily mimicked with an iconic display, there are a number of application areas where there is no obvious visual representation (O'Keefe and Davies, 1986a). A prime example of this is the simulation of real-time software.

Despite the recent and continued increase in the quality of visual output being used, there is no evidence that this increase results in a proportionate increase in model credibility or quality of decision.

(5) *New Methods for Statistical Analysis are Needed.*

One view has it that once the display has been used as a communication medium, it should be turned off, and statistical results obtained as normal. However, *if* the user is consistent (admittedly, this may not always be the case), then user interaction can be just another variable factor that needs to be incorporated into the analysis. Appropriate statistical techniques need to be developed.

4. IS THERE A METHODOLOGY FOR DOING IT RIGHT ?

Differences in the perceived benefits of VIS, and the position taken by some of the folklore, can be put down to the fact that practitioners and researchers are actually using a number of different methodologies for supporting decision making. Many perceptions are a function of the particular paradigm carried by the perspective VIS developer. There are at least five competing methodologies :-

(1) *Statistical* or *Traditional*. VIS is a *selling* aid, which is an adjunct to the normal simulation method. Decision choices are still supported by statistical experimentation: at best, the user can interactively choose from a number of options generated by analysis. There is no facility to interactively generate or explore new options.

(2) *Decision Support*. VIS is a decision support method, helping the user to interactively solve semi or ill-structured problems. The user has complete freedom to apply the model as he thinks fit.

(3) *Computer Aided Design (CAD)*. VIS is used to design a system by coupling together pre-defined parts, much as when CAD is used to design, for example, a complex electronics device.

(4) *Gaming*. VIS is closer in conception to gaming than to discrete simulation. The user benefits from the VIS by learning, not by analysis of results.

(5) *Simulator*. A VIS is a simple simulator, where the user is a 'man-in-the-loop'.

The traditional methodology is prevalent in the USA amongst users of animation, whereas the very different decision support methodology is prevalent in the United Kingdom. The CAD methodology is common amongst users of VIS in manufacturing systems design. Both the gaming and simulator methodologies are not so common, but examples of these views have appeared, for instance Hollocks (1984) and Spearman (1980) respectively.

Any specific approach to visual output, user interaction or visual input, as discussed earlier, or the employment of a certain approach to software packaging, has a considerable influence on the resulting VIS system. For example, with an object-oriented approach, where new instances of objects can be generated by users (for example, SimKit), the CAD methodology is natural. When an animation is post-processed, anything other than the traditional methodology is difficult to achieve. If a package provides a set of standard interactions, then its use under the decision support methodology may be limited if the available interactions do not meet the users requirement to generate new options.

None of these methodologies are necessarily correct or incorrect. Each provides a useful indicator of application. However, *each of these on its own is inadequate* for describing the entire capabilities of VIS. What is needed is a single unifying methodology, built upon elements of all five of these, taking account, and replacing, the folklore discussed above. Such a methodology could have great benefits. At the least, practitioners and researchers will be able to point to a common body of knowledge that is not camouflaged by simple syntactic differences. At best, a methodology would provide :-

(1) A basis for knowing when VIS is appropriate and when it is inappropriate. Typically, practitioners that perceive a single methodology only think that VIS is appropriate when the particular methodology is appropriate.

(2) A method for its proper use, allowing VIS to be any of these methodologies as appropriate. For example, a simulation developed under the animation methodology should not be constrained (by use of software, approach to design, etc.) from evolving into a broader based decision support VIS.

(3) A basis for developing future VIS packages. Too many packages have one of the particular methodologies inherent. For example, WITNESS employs the decision support approach, CINEMA the animation approach, and AutoMod the CAD approach.

(4) A common consensus on future research.

A single methodology may evolve from work in the broad area of Visual Interactive Modelling (VIM). (In a Visual Interactive model, the underlying model may be mathematical or heuristic, rather than a simulation model.) Some discussions on the broader perspective of VIM methodology can be found in Bell (1985).

1.5 CONCLUSIONS

Confusion about VIS follows from differing views about decision aiding *methodology*. Those doing VIS are employing at least five different methodologies. Packages developers are, often inadvertently, providing developers with an inherent decision aiding methodology.

There is no single methodology for doing it right; any might be appropriate in a particular situation, but each is inadequate to describe the entire process. If VIS is going to develop, a single unifying methodology is needed.

ACKNOWLEDGEMENTS

Some of the 'competing methodologies' where first recognized by Peter Bell.

REFERENCES

Descriptions of CINEMA, XCELL, WITNESS and AutoMod can be found in the Software Tutorials section of the *Proceedings of the 1986 Winter Simulation Conference*, and probably in these Proceedings as well.

A.P. Amiry (1965) The simulation of information flow in a steelmaking plant, in *Digital Simulation in Operational Research* (Hollingdale, ed.) English Universities Press, London, England, 347-356.

P.C. Bell (1985) Visual Interactive Modelling in Operational Research: successes and opportunities, *Journal of the Operational Research Society* **36**, 975-982.

P.C. Bell and R.M. O'Keefe (1987) Visual Interactive Simulation - history, recent developments and major issues, forthcoming in *Simulation* **48**.

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

E. Fiddy, J.G. Bright and R.D. Hurrion (1981) SEE-WHY: interactive simulation on the screen, in *Proceedings of the Institute of Mechanical Engineers* **c293-81**, 167-172.

W.S. Faught (1986) Applications of AI in engineering, *Computer* **19**:7, 17- 27.

A. Goldberg and D. Robson (1983) *SMALLTALK-80: The Language and its Implementation.* Addison-Wesley, Reading, MA.

B. Hollocks (1983) Simulation and the micro, *Journal of the Operational Research Society* **34**, 331-343.

R.D. Hurrion and R.J.R. Secker (1978) Visual Interactive Simulation an aid to decision making, *Omega* **6**, 419-426.

IEEE (1984) *Proceedings of the IEEE Computer Society Workshop on Visual Languages*, IEEE, Piscataway, NJ.

Insight International Limited (1986) OPTIK Software Description, 21B Oxford Street, Woodstock, Oxon OX7 1TH, UK.

J.B. Macintosh, R.W. Hawkins and C.J. Shepard (1984) Simulation on microcomputers - the development of a Visual Interactive Modelling philosophy, in *Proceedings of the 1984 Winter Simulation Conference* (Sheppard, Pooch and Pegden, eds.), IEEE, Piscataway, NJ.

B. Melamed and R.J.T. Morris (1985) Visual Simulation: the Performance Analysis Workstation, *Computer* **18**, 87-94.

R.M. O'Keefe (1987) An interactive simulation description interpreter, *Computers & Operations Research* **14**, 273-283.

R.M. O'Keefe and R.M. Davies (1986a) A microcomputer system for simulation modelling, *European Journal of Operational Research* **24**, 23- 29.

R.M. O'Keefe and R.M. Davies (1986b) Visual discrete simulation with Pascal_SIM, in *Proceedings of the 1986 Winter Simulation Conference* (Wilson, Henriksen and Roberts, eds.), IEEE, Piscataway, NJ, 517-525.

C.R. Standridge (1985) Performing simulation projects with The Extended Simulation System (TESS), *Simulation* **45**, 283-291.

M.L. Spearman (1980) Dynamic interactive system simulation: an inexpensive approach to solving logistics problems, in *Proceedings of the Spring Annual Conference*, American Institue of Industrial Engineers.

AUTHOR BIOGRAPHY

BOB O'KEEFE is an associate professor in the Department of Decision Sciences and Engineering Systems at Rensselaer Polytechnic Institute, Troy, New York. He was previously a member of the Board of Studies in Management Science at the University of Kent at Canterbury, England and visiting assistant professor in the Department of Computer Science at Virginia Tech. He received a B.Sc. in Computer Studies and Operational Research from the University of Lancaster (England) in 1979, and a Ph.D. in Operational Research from the University of Southampton (England) in 1984. Research interests include Artificial Intelligence and simulation, Visual Interactive Simulation, the application of expert systems, and qualitative approaches to decision making. He is a member of SCS, TIMS, ORS, AAAI and BCS, a Director of Decision Computing Limited, and has previously published in the *Journal of the Operational Research Society*, the *European Journal of Operational Research*, the *Computer Journal*, *Simulation*, *Computers & Operations Research*, and *IEEE Expert*.

Robert M. O'Keefe
Department of Decision Sciences and Engineering Systems
Rensselaer Polytechnic Institute
Troy, NY 12180-3590, USA