

A UNIFIED FRAMEWORK FOR VISUAL INTERACTIVE SIMULATION

Michael Rooks

WATMIMS Research Group - Department of Management Sciences
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

ABSTRACT

Visual interactive simulation (VIS) is gaining widespread popularity among simulation practitioners. Although interest in research and development grows at a corresponding rate, researchers have yet to agree upon the general objectives, requirements, and capabilities of a system for true visual interactive simulation. Ambiguous terminology in the literature makes it difficult for the user community to understand the potential capabilities and limitations of VIS. This paper outlines general requirements of VIS systems, to allow the development of a unified framework for VIS. An application-independent terminology, which seeks to clarify and organise the concepts of VIS, is proposed. VIS requirements are then related to implementation issues, to provide practitioners with an understanding of the scope of alternatives for enhanced model analysis using visual interactive simulation.

1 INTRODUCTION

Visual interactive modelling (VIM) and visual interactive simulation (VIS) are rapidly emerging as powerful tools for the development and analysis of large and complex dynamic systems. The promise of computerised visualisation is the ability to concentrate data into concise forms which emphasize some informational aspects of the data, and de-emphasize others. By highlighting important relationships, trends, values, or logic, the visual display can effectively deliver the otherwise unclear messages conveyed by raw data. The promise of interaction is the ability of the analyst to work directly with a model, influencing its behaviour, and experimenting with its dynamical properties and relationships.

The 1980's witnessed the emergence of VIM and VIS as viable, cost-effective tools for enhancing the

processes of modelling and analysis. More recently, researchers have begun to refine the terminology associated with VIM and VIS; however, some contradictions and ambiguities in terminology remain. These deficiencies prevent the user community from effectively expressing its needs, depriving simulation software developers of the feedback required to provide for these needs. Moreover, this lack of clarity has resulted in redundancy in some areas of research in the field, and inadequate attention to others. While the construction of models in a visual, interactive environment has been the subject of concentrated research efforts, the processes of interactive experimentation and analysis have received little direct attention.

A broad, application-independent terminology for the field of VIM/VIS, if communicated simply and concisely, can help to alleviate communication problems between academics, system developers, and the user community. In addition, it can lay the foundation for discussion and research which recognises and encompasses the broader implications of VIM/VIS. Valuable insights into visual interactive simulation, (the process of interactively experimenting on a computer model of a dynamic system) can be gained through a careful review of relevant terms and concepts.

This paper aims to encourage and facilitate dialogue between the various simulation communities, by refining the definitions of VIS and its components, and presenting them in a way which is illuminating to the researcher, instructive to the developer, and comprehensible to the user. In the process, several myths about the capabilities and future potential of VIS are dispelled. It is shown that the central components of VIS are application-independent, giving the discussion and resulting terminology relevance across the spectrum of simulation applications.

An exploration of the overall goals and requirements of VIS is then undertaken, utilizing the new concepts and definitions. This exploration suggests new alternatives for the design of more effective environments for VIS. These alternatives are evaluated, and useful avenues for future research into VIS are suggested.

2 BACKGROUND AND GENERAL DEFINITIONS

Techniques for visual interactive modelling and simulation are rapidly becoming necessary components of most simulation studies. The potential benefits [Bell & O'Keefe, 1987; Hurrion, 1986] and pitfalls [Paul, 1989] of these techniques are well-documented. However, the literature has not yet provided consistent, unifying methodologies for describing or implementing either VIM or VIS [O'Keefe, 1987].

In order to provide a unifying framework for VIS, it is first necessary to define some concepts fundamental to the simulation activity. The terminology used to describe simulation concepts has often been contradictory, largely due to a lack of distinction between a *model*, and the processes which act on the model (see for example, Bishop & Balci [1990]). Models are constructed and analysed through the processes of *modelling* and *simulation*, respectively.

The following definitions serve to resolve the contradictions and ambiguities which arise when the object (model) and the procedures (modelling, simulation) are not properly defined [Rooks, 1990]. *Computer modelling* is the process of representing a real or imaginary system using a combination of data and programs on a computer. When the computer is used to model dynamic systems which change over time, a *simulation model* must be developed. *Simulation models* are therefore computer models of dynamic systems, and *simulation modelling* is the process of constructing a computer model of a dynamic system. *Computer simulation* (or simply *simulation*) is the process of using a simulation model to test hypotheses about a modelled system. Thus, *simulation* is the act of experimentation with a computer model of a dynamic system. Finally, a *simulation study* describes the processes of modelling and simulation in the broadest sense, and may encompass the entire life-cycle of a simulation modelling project [Balci, 1989].

The definitions provided above allow VIM and VIS to be defined accurately and unambiguously. *Visual interactive modelling* is the process of creating a computer model with the aid of a graphical

programming environment. *Visual interactive simulation*, on the other hand, is the process of model experimentation and analysis with the aid of a graphical environment for interaction between the model and the analyst. These definitions imply that VIM focusses on the programmer of the simulation model, while VIS focusses on the user of the model. In the past, the VIS label has been applied to either VIM or VIS as defined above, or both. The confusion resulting from an ambiguous definition has caused research on "VIS" (quotation marks denote a more general definition which includes VIS and VIM) to take several divergent paths.

The literature on "VIS" may be roughly divided into four categories:

Specification and Description of Systems for Visual Interactive Modelling. This research focusses on the use of graphical tools for model specification. The resulting tools typically provide a graphical language which allows the modeller to construct networks or scale diagrams of the system under study. Additional detail is then supplied via data entry, and the graphical model is translated into source or executable model code. Recent examples of this type of work include [Bishop & Balci, 1990; Korn, 1989; Richmond *et al*, 1987]. Several of these tools claim to provide for VIM and VIS. Often, however, application-specific VIM becomes the primary research focus, and practical, general VIS techniques are neglected.

Tutorials for the VIS Novice. These articles present VIS concepts, and the various alternatives for their implementation, in a form which is comprehensible to a new or prospective user of VIS [Hurrion, 1989; Bell & O'Keefe, 1987]. Unfortunately, these articles often provide a narrow view of VIS, providing a better indication of currently available technologies than of technical possibilities. For example, such an article may consider a system which provides animation to represent VIS [Hurrion, 1986]. While providing a useful description of animation techniques, it may leave the reader with an incomplete understanding of the many diverse techniques for VIS.

Reviews of VIS Benefits and Pitfalls. This research strives to clarify the potential capabilities of VIS, and to point out the possible pitfalls which may result from the use of VIS techniques. Without a clear definition, however, the potential of VIS is not easily quantified. Without a unified approach to VIS, it is not possible to consider the facilities which VIS systems may provide to alleviate its potential pitfalls.

Future Prospects of VIS. These articles were written primarily when VIS was in its infancy [Henriksen, 1983; Ören & Zeigler, 1979]. They provide valuable benchmarks against which actual progress may be compared, and examine the requirements of VIS in more detail than most current articles on VIM/VIS research.

The reader is referred to the articles listed above for further information on these "VIS" topics. Collectively, the literature provides an excellent source of information on VIS and VIM; however, contradictions and ambiguities in terminology are apparent within and among them. The remainder of the present paper describes VIS in terms of the processes for interaction between the simulation analyst and the executable model, during the experimental and analytical phases of a simulation study.

3 REQUIREMENTS OF A SYSTEM FOR VISUAL INTERACTIVE SIMULATION

Techniques associated with VIS are invaluable in many simulation studies. It is instructive to consider first *why* they are desirable, and often necessary, for the successful execution of a simulation study, before discussing *how* specific VIS techniques can enhance the simulation process. As the power of simulation hardware and software increases, models of systems exhibiting formidable complexity are being developed. In order to cope with this complexity, model analysts are demanding tools which help to organise, refine, and present information derived from the execution of a model, in forms conducive to understanding the dynamics of the modelled system. The primary requirement of VIS is to provide tools to meet this need. VIS systems must aid in qualitative understanding by providing insight into system dynamics, and they must support quantitative verification of these insights. VIS systems must also promote efficiency in the processes of model verification and experimentation.

Secondary effects of model complexity also make VIS techniques invaluable. At the outset of model analysis, the analyst may not understand system dynamics sufficiently to plan an effective strategy for experimentation. VIS must allow the modeller to qualitatively and quantitatively explore system dynamics both prior to, and during, statistical experimentation. Additionally, the study of complex systems often involves long timeframes, causing model parameters to become obsolete during the course of

a model run [Stigliani *et al*, 1989; Gault *et al*, 1987]. Such long-term planning models demand that VIS provide capabilities for intervention during model execution. Finally, as a larger and more diverse body of simulation users develops, modelling objectives also become more diverse [Stigliani *et al*, 1989]. Models are being used with increased frequency in educational programs, and for such real-time applications as operator training and on-line process control.

There are many additional reasons for the emergence of VIS tools, and many ways in which VIS may aid in solving problems [O'Keefe, 1987]. However, a unifying approach to VIS must apply a broad perspective, in order that the fundamental requirements of VIS can be identified. Such a perspective recognises that all models are inherently similar, in that they provide a logical component, in the form of model code, and a data component. As such, there are identifiable modes of interaction which occur between the user, the model, and the simulation system. Four fundamental requirements of VIS may be identified which apply to any computer model of a dynamic system, throughout the simulation process. Each is a necessary condition for VIS, and all are collectively sufficient to convert a simulation system into a complete system for VIS [Rooks, 1990]. These requirements apply to *at least* all model parameters under experimentation; they may apply to model code as well, as determined by modelling objectives. The four requirements for a complete VIS system are:

- | | |
|----------------------|--|
| <i>Intervention</i> | The analyst must be provided with an effective means of initiating interaction with the model. Modes of interaction include inspection, specification, and visualisation. |
| <i>Inspection</i> | The analyst must have access to all model data relevant to the experiments to be performed on the model. The nature of this access (eg. read only, or read/write) is determined by the objectives for analysis of the model. |
| <i>Specification</i> | The analyst must be capable of specifying model parameters, in accordance with the objectives for analysis of the model. |
| <i>Visualisation</i> | The analyst must be capable of viewing model data in ways which illustrate the model dynamics and |

relationships of interest. In order to accommodate various models and modelling objectives, capabilities for visualisation must be diverse, flexible and parametric.

4 VIS CONCEPTS AND DEFINITIONS REFINED

4.1 User Intervention in Model Execution

Intervention by the analyst during the execution of a simulation model is an essential part of VIS [Bell & O'Keefe, 1987]. Henriksen [1983] acknowledges the need to encourage user interaction through model feedback. Without capabilities for intervention, a simulation system is not truly interactive, since the user cannot experiment with model parameters during execution. Bishop & Balci [1990, p. 505] describe two modes of intervention:

Model prompted interaction occurs when the simulation itself prompts the user for input. *User prompted interaction* is characterized by the user's ability to specify when the interaction is to occur.

This description is ambiguous for several reasons. The concept of intervention has been confused with interaction, leading the reader to believe, for example, that simply viewing an animation is not a form of interaction. Additionally, the user does not need to prompt for interaction; rather, he/she simply *initiates* interaction by executing various commands. Finally, the use of *simulation* in the description leaves the prompter of the interaction ambiguous; simulation is a process, not an object.

User intervention in model execution can be initiated in three ways. *User-initiated* intervention occurs when the user issues a command with an input device, which causes a change in the state of the model or simulation system, or allows further input from the user. *Model-initiated* intervention occurs when the state of the model reaches a condition which requires information from the user. *System-initiated* intervention occurs when the state of the VIS system reaches a condition which requires information from the user. For example, if windows are open on overlapping dynamic displays, the VIS system may require information from the user to resolve display conflicts.

Effective intervention requires the user to detect the correct time to make changes to model data, and to make the necessary changes to model data when such a time arrives. Interventions initiated by the model or

VIS system determine the correct time. The timing of user-initiated intervention is determined by the state of the model, as communicated by dynamic displays. A dynamic display of simulated time should be provided by the VIS system for this purpose. Specification of changes to model data requires intelligent packaging of the VIS tools for inspection, specification, and visualisation of model parameters which are described in Sections 4.2 to 4.4.

More complex functions related to user intervention may also be required. Often, decisions on design, policy, or management issues must be played against one another, to determine those which generate optimal system responses. Such decisions are often made at a critical point in model execution; it is therefore necessary to provide the capability to save a complete description of the model state at such a point in simulated time, to allow various interventions to occur from that time. In general, it is desirable to provide the capability for saving any number of states, so that the model can be reset to any pre-determined time in its execution history.

As noted in Section 2, difficulties related to statistical validity may be introduced by a naive user of VIS intervention capabilities [Paul, 1989]. Many of these may be alleviated by a well-designed system for user intervention. For example, it is extremely important to maintain a record of all interventions, in order that model output can be reconciled with all inputs to model parameters specified by the user. Additionally, the VIS system should provide modes of execution which warn the user or prevent the re-specification of model variables during multiple replications of a stochastic model.

Capabilities for intervention in model execution vary among VIS systems. Most provide standard "function" keys, which allow pre-programmed interaction functions to be executed, for example: specification of graphical window parameters, or modification of data for model replication control and statistical functions. Some provide for the use of a pointing device as an alternate or additional means of interaction. Systems which meet all requirements of true VIS provide additional capabilities for accessing model-specific parameters, and for the execution of model-specific functions which are programmed by the modeller.

4.2 Data Specification and Inspection

Several methods can be used to provide a user-friendly interface for data specification and inspection. The most effective methods are those which make the

organisation of the data natural and easy for the user to comprehend. In simulation modelling, the simplest means of achieving a natural data interface is through the design of a data specification system which uses the same hierarchical structure as the system under study. By organising data in a hierarchy corresponding to that of the modelled system, the analyst's understanding of the organisational framework is assured. An example of an effective scheme for organising simulation data is a hierarchical menu system, which utilizes data tables for storing model data [Rooks, 1991; Rooks, 1990]. Menus provide access to all data tables in the hierarchy, while tables provide a natural interface for data inspection. Data specification and manipulation may also be carried out using a set of operations defined for the table. These may include console input functions, copying from one table to another, or cutting and pasting data within a table. A well-known example of a multiple-data-type table is the ubiquitous spreadsheet metaphor.

4.3 Visualisation Concepts

Tools for visualisation are arguably the most difficult components of VIS to design and implement; however, visualisation tools can determine the effectiveness of the entire VIS interface. There has been a tendency in the literature and research on simulation and modelling, to take a narrow view of general visualisation concepts. This tendency is likely due to the practical, applied nature of the simulation activity, and has resulted in a terminology which is very restrictive. The design of graphical displays should not begin with a specific simulation application in mind. Rather, the design of effective tools for the visualisation of simulated systems requires (and benefits significantly from) investigation of some fundamental concepts of graphic design.

VIS systems may provide several graphical displays on the screen at any given time. A *display* is defined as a single instance of any type of graphical image which the VIS system is capable of generating. Several authors have proposed classification schemes for VIS displays; however, these schemes have either been restricted to particular applications, or not sufficiently descriptive to be unambiguous. A new classification scheme is presented below, and is compared with those proposed previously.

VIS displays possess two fundamental properties. The first pertains to the way in which model data is represented [Rooks, 1990; Tufte, 1983], and divides displays into two general families:

Representative displays, in which the system under study is viewed in some simplified form(s) of its real appearance. Representative displays provide familiar visual cues from the system under study to aid in the analyst's comprehension of system dynamics. An animated layout of a manufacturing facility is a popular example.

Abstract displays, in which model data is displayed in alternate form(s) to enhance its comprehensibility. Abstract displays organise and filter data, to ensure that data irrelevant to a given aspect of system dynamics does not cloud the analyst's comprehension. Graphical plots of variable values changing over time, and gauges and meters are examples of abstract displays.

The second property is simply the distinction between *dynamic* displays, which change over simulated time, and *static* displays, which are instantaneous snapshots of aspects of the modelled system.

Representative displays provide a pictorial view of the modelled system, often with a one-to-one correspondence between entities or processes in the system, and those on the screen. Ideally, they provide a range of perspectives on system operation which covers the spectrum from global views to specific, local views, and allow qualitative analysis and verification of multiple dynamic relationships. Representative displays seldom provide an exact representation of the system. They are not necessarily drawn to scale, but in general convey spatial or comparative relationships. They may use symbols or icons to represent entities, and omit irrelevant details to simplify the display. Representative displays may also be used for static examination of data. An example of this is the display of a factory layout which is used for verification prior to running a simulation.

Two general classes of representative displays can be defined: *scale models*, and *schematics*. These classes represent the ends of the spectrum of representative displays; many fall within this spectrum. For example, a visual display of a factory floor may use scale dimensions for physical plant, but indicate system status with schematic aids such as icons. A challenge for research in visualisation is to merge capabilities for scale modelling and schematics into a single conceptual framework.

Several terms have been suggested to describe all representative displays. *Animation* is the most prevalent, and is often considered analogous to VIS [Bell & O'Keefe, 1987; Hurrion, 1986]. However, animation implies a dynamic display. It is important

to recognize that representative displays may be either static or dynamic, and that a flexible system for VIS should provide a common set of constructs for producing both static and dynamic displays. The word *schematic* has recently been suggested in the literature [Bishop & Balci, 1990; Hurrion, 1989], and is an improvement on *animation*. However, *schematic* implies that the display is diagrammatic, or an abstraction from the way the system appears in reality. If one considers photo-realistic images or highly detailed 3-dimensional scale layouts as examples, it is clear that the word *schematic* is too restrictive to describe all representative displays.

Abstract displays are visual tools for data abstraction. They contain data which may have no visual analogue nor any conceptually familiar visual representation. Abstract displays provide essentially fixed-format transfer functions which organise and convert model data values into a comprehensible visual image. The characteristics of the data being viewed, and the objectives of the analyst in viewing it, determine the appearance and functionality of the abstract displays which are required for analysis. The application-independent nature of abstract displays makes them highly desirable from the modeller's standpoint, since they require no programming. Abstract displays are provided by the VIS system, requiring the analyst only to supply parameters and to enable the required displays.

The utility of the abstract display lies in its ability to convert intractable or incoherent data into forms which are clear, concise, and informative. Abstract displays tend to focus more sharply on fewer model variables than representative displays, providing insight into trends or trajectories in system variables. Often, relationships (qualitative and quantitative) between time-dependent or inter-dependent variables may be observed over simulated time, with the abstract display being used to filter out all information irrelevant to the evaluation of such relationships. Abstract displays are often used to summarise large quantities of data by presenting aggregate combinations of raw model data. The literature on VIS largely neglects the uses of dynamic abstract displays; in fact, Bell and O'Keefe [1987] do not mention their use, despite the fact that a dynamic abstract display can provide the clarity of a representative display, and the accuracy of numerical data, without requiring any programming.

Several terms for the description of abstract displays have been proposed and used in the literature. However, a lack of consensus as to the meaning and scope of these terms has motivated the proposal of

new terms in the present paper. In their summary of the state of the VIS art, Bell & O'Keefe [1987] describe abstract displays as *representational*. This term is too vague to preclude ambiguity, since virtually all computer-generated displays are representational to some extent. *Logical* was suggested by Hurrion [1989], and subsequently used by Bishop & Balci [1990]. This term implies that model logic, or a logical relationship between model variables, is presented. In general, abstract displays do not illustrate logical relationships. Bishop and Balci [1990] also use the term *summary* to describe abstract displays. Indeed, many abstract displays summarise data; however, many do not, indicating that the *summary* display is simply a subset of the family of abstract displays.

A challenge posed to VIS researchers and software developers lies in the design of effective and flexible abstract displays. Abstract displays use symbols to illustrate and summarise data and relationships. The spectra of symbols and options for their presentation are virtually unlimited, and to some extent, determine the effectiveness of displays. As demonstrated by the abstract display libraries of common spreadsheet packages, these visual tools are capable of effectively presenting data generated by virtually any modelled system. Abstract displays for VIS are more complicated, however, in that they must function equally effectively in a dynamic mode. Like their representative counterparts, code for abstract displays should provide a common set of constructs for producing both static and dynamic displays. Identification of a standard, general set of abstract display models, which can be effective for both static and dynamic application, requires further research, as does investigation into the problem of making these displays flexible through parametric specification.

4.4 Implementation of Visualisation Displays

4.4.1 VIS Demands a Windowing System

The basic terminology of displays for model visualisation, presented in Section 4.2, can be used to systematically enumerate and evaluate alternatives for the design of visualisation tools. General aspects of visualisation tools, which pertain to both representative and abstract displays, are first discussed. More detailed concepts are then presented for each type of display, and are used to generate alternatives for implementation. Alternatives are described, and their strengths and weaknesses evaluated.

In Section 4.3, it was observed that both

representative and abstract displays may be required to operate in either a static or dynamic mode. Such a capability is automatically provided, if the VIS developer considers the mechanics of generating displays for visualisation in a very general sense. Each display may involve static objects, which present a constant appearance during model execution and display updates, and dynamic objects, which undergo some sort of transformation (eg. translation, rotation, change in color) as model execution progresses. The contents of the display at any given time include the static and dynamic objects which provide a partial description of the state of the modelled system. A static display is simply one frame of an ongoing dynamic display - a snapshot of the dynamic display. It may be generated by the same procedures which initialise a dynamic display (procedures for the generation of static and dynamic object images), and as such, should demand no extra effort in its implementation. Thus, the problem is reduced to the investigation and implementation of dynamic displays, provided that the static and dynamic components are separable.

A simulation system is not like a word-processor, for which an efficient graphical layout can be specified, implemented, and used for long periods of time without modification. In simulation modelling, as experimentation progresses, as analysis requirements change, and as the modelled system changes, the simulation analyst's interface may need to change in order to provide the information in a useful format. Thus, the simulation interface must be flexible, making no assumptions about the analyst's requirements for interaction with the model, and allowing the specification of all aspects of the interactive environment. The VIS system would ideally provide the analyst with capabilities for:

- 1) Interactive parametric definition of display windows.
- 2) Interactive parametric description of a comprehensive set of display types for data visualisation, inspection, and specification.

A window-based system for model visualisation serves several purposes. A display window may be interactively opened, closed, copied, moved, and resized, without the need to change the description of window contents. This eliminates the need to define a *null* display type [Bishop & Balci, 1990] which confuses the distinction between an active/inactive display and a defined/undefined display. A window-

based system allows several displays and display windows to be defined, and uses a flag to determine whether a given window is active, and a pointer to the display which it contains. Window-based VIS systems allow multiple instances of a given display to be presented. For example, code which generates a two-dimensional plant layout could be used for several displays simultaneously, perhaps global and zoomed views. Conversely, the contents of a display window may be changed while the size and location of its window remains constant. For example, several layouts may be swapped in and out of the display window as required. The alternatives for abstract and representative displays described below assume that a window-based system is used. Thus, the discussion focusses on the *content* of various display types, rather than the *mechanics* of managing them on the display screen.

4.4.2 Implementation of Representative Displays

Section 4.2 indicated that representative displays fall into a spectrum which is bounded by scale models with a high degree of realism, and schematics, with symbolic representations. There are several alternatives for the implementation of VIS tools which address parts of this spectrum. In general, modern tools for representative visualisation utilise one of two implementation techniques. Those which generate and manipulate displays based on screen coordinates (either in text or graphics mode) are referred to as *iconic*. Icons may include bit-mapped images (*pixel-based*) or characters (*character-based*). Techniques which transform system coordinates automatically into screen coordinates are referred to as *geometric*. In general, schematics are most efficiently generated with iconic displays, while scale models are best generated using geometric displays.

Pixel-based and character-based visualisation systems exhibit many common properties. Static objects in the display are typically described using drawing packages which create a representation of the modelled system based on screen coordinates. Dynamic objects are represented by icons. In character-based systems, objects are represented by displayable characters (possibly defined by the modeller), while in pixel-based systems, a facility is typically provided for defining the appearance of icons, and the creation of an icon library. Dynamics are indicated by applying transformations to icons, such as translation, rotation, and color change. Iconic displays can only provide very primitive facilities for transforming their appearance. Layouts which are

larger than the display screen may be panned, but capabilities for zooming the layout are restricted to magnification. The magnification technique does not provide increased detail, which is usually the desired result of a zoom operation. Iconic displays are typically two-dimensional, although bit-mapped images may be drawn from an oblique perspective to provide the illusion of three dimensions.

Geometric visualisation systems transform the geometry of physical objects in a model into drawing commands which generate a display of some aspects of the model. The computational effort expended in converting raw geometric data into transformed screen coordinates is considerably greater than that required to update an iconic display; however, the advantages are numerous, and trends in computing hardware performance indicate that the computational aspect is of quickly-diminishing importance. Geometric visualisation systems provide libraries for all common activities associated with the generation and windowing of displays. However, due to the flexibility and detail which geometric displays can provide, they typically require more model-specific programming than iconic displays. Geometric displays support an object-oriented approach to modelling, in that visualisation modules are driven by the same data as the modules which control model logic. The resulting implications for flexibility in model experimentation are significant. For example, when a new plant layout is specified, the visualisation facility can automatically utilize the new data, requiring no re-programming. Transformations to displays can be made interactively, by modifying such parameters as viewpoint, magnification, or the limits on the display specified in the coordinates of the modelled system. With a well-designed windowing system, several views of a system may be defined, and selected as required for display in one or more windows. This allows, for example, a global view and several detailed views of the modelled system to be displayed simultaneously. Geometric systems have been developed for the display of data in two or three spatial dimensions [Brunner & Henriksen, 1989; AutoSimulations, 1989].

4.4.3 Implementation of Abstract Displays

The design or selection of a VIS system involves many issues associated with provisions for abstract visualisation. Decisions must be made as to the types of displays which are provided and the flexibility of displays in meeting varied analysis needs. The designer of a library for abstract VIS displays must select from a large range of display types which have

been suggested and implemented in software packages for simulation and data analysis. Additionally, it may be necessary or desirable to define new display types, possibly for a specific application, or for general use. The simulation analyst must select a package which meets the requirements of his/her specific application.

An abstract display can assume virtually any form which effectively conveys information to the model analyst. Two primary properties distinguish the various abstract displays, and provide a basis for evaluation of their appropriateness in a given VIS system or application. In general, abstract displays may be *cumulative*, providing increasing amounts of data and insight into system operation as execution progresses, or they may be *instantaneous*, reflecting only the current state of selected system variables. Additionally, the data which is used by an abstract display may be *bounded* by limitations imposed by the display type or by the feasible range of model data, or it may be *unbounded*, requiring the display to accommodate data of virtually any magnitude.

Cumulative abstract displays serve two primary purposes in the simulation context: to observe model variables over time, or to observe model variables in relation to each other. The most common application is a graph of one or more variables as a function of time. Selected variables may be updated at a pre-determined interval in simulated time, or they may be sampled upon occurrence of an event. In either case, they are plotted as a function of the independent time variable, with previous values remaining on the display until space restrictions force them to be removed. Cumulative displays also include the more general x-y plots. These may take the form of line graphs or scatter plots, with data points typically remaining on the screen for the duration of the model run.

Instantaneous abstract displays serve many functions, and are more flexible than their cumulative counterparts because they are not required to retain any model data. Instantaneous displays are completely updated at each interval, transforming selected data values into graphical images, usually for comparison or evaluation of their magnitude. Standard instantaneous displays include bar charts, histograms, and pie charts. Other possibilities include box plots [Tukey, 1977], and an assortment of gauges and metres, such as dials and level indicators. They may appear in a digital or analogue form, and their data may be displayed in absolute terms, or normalised, perhaps as percentages.

The data table, discussed as an example of a medium for data inspection and specification in

[Rooks, 1990], is also an example of an instantaneous abstract display. The windowing system must allow any model data table display to be opened, to provide the capability of data inspection and specification during model execution.

Displays which are bounded typically maintain a fixed appearance, with indicators of data values scaled to fit within the window to which the display is assigned. Parameters such as the number of variables, and the range of data values are used to customise the display for different applications.

Unbounded displays are necessary for many applications which require the relationships between variables of different orders of magnitude to be observed. Unbounded displays accommodate widely-varying data values in several ways. A data interval of interest may be parametrically specified, allowing the display to be scaled to accommodate the interval. Variables which fall inside this interval are plotted, while those falling outside are ignored. Alternatively, the display may be rescaled under certain conditions, to accommodate variations in data values. Finally, digital displays and exponent multipliers can allow variables to be plotted within the display window.

5 SUMMARY

Objectives, capabilities, requirements, and concepts related to systems for visual interactive simulation have been discussed. Definitions of VIS and VIM, which are derived from precise definitions of computer modelling and simulation, have been proposed. The definition of VIS, and the objectives of VIS techniques, determine the requirements for four essential capabilities of a VIS system: intervention, inspection, specification, and visualisation.

VIS requirements have been defined and described individually. Methods of intervention were defined rigorously, and several issues related to VIS intervention capabilities were discussed. Displays for visualisation were then shown to fall into two families: representative and abstract. The representative family was further divided into scale models and schematics, while abstract displays were subdivided into cumulative and instantaneous types. It was additionally shown that all displays should be capable of presenting similar data in static or dynamic forms, if properly implemented. Issues in the implementation and use of each class and type of display were described, introducing the concept of bounded and unbounded abstract displays.

The consistent and complete terminology presented

herein is proposed as a general vocabulary upon which dialogue on VIS may be based. It is applicable to all VIS applications, and promotes effective communication within and among the various simulation communities.

REFERENCES

- AutoSimulations, Inc. 1989. *Automod II User's Manual*. Bountiful, Utah.
- Balci, O. 1989. How to assess the acceptability and credibility of simulation results. In *Proceedings of the 1989 Winter Simulation Conference*, eds. E.A. McNair, K.J. Musselman, and P. Heidelberger, 62-71. San Diego, CA: Society for Computer Simulation.
- Bell, P.C., and R.M. O'Keefe. 1987. Visual interactive modelling - history, recent developments, and major issues. *Simulation* 49:3, 109-116.
- Bishop, J.L., and O. Balci. 1990. General purpose visual simulation system: a functional description. In *Proceedings of the 1990 Winter Simulation Conference*, eds. O. Balci, R.P. Sadowski, and R.E. Nance, 504-512. San Diego, CA: Society for Computer Simulation.
- Brunner, D.T., and J.O. Henriksen. 1989. A general purpose animator. In *Proceedings of the 1989 Winter Simulation Conference*, eds. E.A. McNair, K.J. Musselman, and P. Heidelberger, 155-163. Society for Computer Simulation: San Diego, CA.
- Gault, F.D., K.E. Hamilton, R.B. Hoffman, and B.C. McInnis. 1987. The design approach to socio-economic modelling. *Futures*, February 1987, 3-25.
- Henrikson, J.O. 1983. The integrated simulation environment. *Operations Research* 31:6, 1053-1073.
- Hurrion, R.D. 1989. Graphics and interaction. In *Computer Modelling for Discrete Simulation*, ed. M. Pidd, 101-119. Toronto: John Wiley & Sons.
- Hurrion, R.D. 1986. Visual interactive modelling. *European Journal of Operational Research* 23: 281-287.
- Korn, G.A. 1989. *Interactive Dynamic System Simulation*. Toronto: McGraw-Hill.
- O'Keefe, R.M. 1987. What is visual interactive simulation? (and is there a methodology for doing it right?). In *Proceedings of the 1987 Winter Simulation Conference*, eds. A. Thesen, H. Grant, and W.D. Kelton, 461-464. San Diego, CA: Society for Computer Simulation.
- Ören, T.I. and Zeigler, B.P. 1979. Concepts for advanced simulation methodologies. *Simulation* 32:3, 69-82.

- Paul, R.J. 1989. Visual simulation: seeing is believing? In *Impacts of Recent Computer Advances on Operations Research*, eds. R. Sharda, B.L. Golden, E. Wasil, O. Balci, and W. Stewart. New York: North-Holland.
- Rooks, M.R. 1991. *The FRAMES Simulation System - User's Manual*. Waterloo, Ontario: Interactive Dynamic Modelling.
- Rooks, M.R. 1990. *An Integrated Framework for Modelling Complex Dynamic Systems*. MASC Thesis, University of Waterloo, Waterloo, Ontario.
- Stigliani, W.M., F.M. Brouwer, R.E. Munn, R.W. Shaw, and M. Antonovsky. 1989. Future environments for Europe: some implications of alternative development paths. *International Institute for Applied Systems Analysis (IIASA), Executive Report 15*. Laxenburg, Austria: IIASA.
- Tufte, E.R. 1983. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press.
- Tukey, J.W. 1977. *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.

AUTHOR BIOGRAPHY

MICHAEL ROOKS is Associate Director of the Waterloo Management of Integrated Manufacturing Systems (WATMIMS) Research Group at the University of Waterloo. After completing his B. Eng (Manufacturing) at McMaster University in 1986, he worked in the automotive industry as an industrial engineer and simulation analyst. His interest in modelling complex systems led him to the University of Waterloo, where he completed an MASC (Systems Design Engineering) and the prototype for a new user-centered system for VIS. Current research interests include: modelling methodology, visual interactive modelling and user-centered simulation technologies, modelling and management of complex interdisciplinary systems, and tools for environmental modelling and decision-making.