Fall 2004

A Template-Based Conceptual Modeling Infrastructure for Simulation of Physical Security Systems

Paul Savory
University of Nebraska at Lincoln, psavory2@gmail.com

Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

A TEMPLATE-BASED CONCEPTUAL MODELING INFRASTRUCTURE FOR SIMULATION OF PHYSICAL SECURITY SYSTEMS

Ashu Guru
Paul Savory

Department of Industrial and Management Systems Engineering
175 Nebraska Hall
University of Nebraska
Lincoln, NE 68588-0518, U.S.A.

ABSTRACT

Although simulation is one of the most innovative and cost-effective tools for modeling and analyzing a system, simulation studies often fail to provide any useful results. One reason is attributed to the fact that *model formulation* depends on the *skills of the analyst*. This paper describes a research to develop a conceptual modeling infrastructure to assist a simulation analyst in specifying components for studying physical security systems. The modeling framework has been programmed as an internet-based web application. Using the application, the successful development and implementation of a physical security simulation model will be aided by a defined scientific methodology rather than simply the skills of the analyst. Further the modeling framework is simulation language independent, thus allowing for a top-down or bottom-up approach to developing the conceptual model. This offers support for an object-oriented modeling design.

1 INTRODUCTION

Since the terrorist attacks of September 11, 2001, the requirements of constantly managing and re-evaluating all direct and indirect risks in physical security systems has increasingly become more important. In order to effectively plan for and manage the operations of such systems, it is essential to constantly analyze its current and future policies, procedures, and equipment. Computer simulation has been proven to be a useful methodology to study business and industrial system behavior under a variety of conditions. It provides a means to analyze the simultaneous interaction of many system variables to yield valuable insight (Rowe 1960). As a result, computer simulation can provide answers in the analysis, planning, and maintenance of physical security systems.

Although simulation is one of the most innovative and cost-effective tools for system modeling and analysis, simulation studies often fail to provide any useful results (Annino and Russell 1979; Keller, Harrell and Leavy 1991; Robinson and Pidd 1998). One reason is attributed to the fact that *model formulation* – a key step in a simulation study – requires an analyst to work from a sense of the problem, envision and assemble the elements, and identify dependencies and relationships that logically comprise the variables of the actual system. Thus, the success of a simulation study is highly dependant on an analyst's domain knowledge, capability to understand the system components, their input parameters, and the interrelationships among those variables and parameters. Reviews on failed simulation studies done by Annino and Russell (1979) and Robinson (1999) highlight that the most common reason for failure is an incomplete mix of essential modeling skills of the analyst. Modeling skill is the ability of an analyst to design a conceptual model that imitates the system under study at the required level of detail. It has been also defined as the skill of the analyst to understand the problem to be tackled and then correctly identify the required modeling parameters and dependent variables.

Willemain's (1995) research on observing how simulation experts formulate problems, found that they spent 59% of their time on structure, 16% on assessment, 14% on context, 9% on realization, and 2% on implementation. Table 1 shows the most time consuming questions that experts address when conducting a simulation study.

Table 1: Most time consuming questions for experts to answer during a simulation study (Willemain 1995)

- What are the (system) variables?
- What are the relationships among the (system) variables?
- What kind of model should I make?
- What process would I follow to make the model?
- How should I analyze the data to understand the problem?
- What are the steps in any model defined as procedure?

In order to answer the questions in Table 1, an analyst needs to understand the physical system, interview employees of the system, and then use her skills to help build the model. Obviously, any tool in helping identifying the dependent variables and their interrelationships in a defined domain will be an invaluable tool in a simulation study. Such a tool will improve efficiency, productivity, quality, and lower the probability of leaving key system elements out of the conceptual model. Development of such a tool seems even more important when there are large numbers of similar simulation studies that are being conducted within a single domain. One such domain is physical security systems, where the number of simulation studies to be conducted will continue to rise. Table 2 is a list of a few recent studies of such systems.

Table 2: Previous works exploring aspects of physical security system

Security System Throughput Modeling (Leone, K. 2002)
Simulation of Check-In at Airports (Joustra and Dijk 2001)
Optimum Design and Operation of Airport Passenger Terminal Building (Saffarzadeh and Braaksma 2000)
Washington Dulles International Airport Passenger Conveyance Study (Kyle 1998)
An Optimum Resource Utilization Plan for Airport Passenger Terminal Building (Parizi and Braaksma, 1995)
Analysis and Simulation of Passenger Flows in an Airport Terminal (Gatersleben. and Weij 1999)
Distributed Real-Time Simulation for Intruder Detection System Analysis (Smith et al. 1999)
Discrete-event Simulation for the Design and Evaluation of Physical Protection Systems (Jordan et al. 1998)

2 INFRASTRUCTURE

This section describes the foundations, the methodology, and infrastructure for creating a conceptual modeling framework for models of physical security systems. Section 2.1 discusses previous efforts in automated simulation model development. Section 2.2 explains the methodology and the infrastructure of this research.

2.1 Previous Work in Automated Model Development

Generic or template-based simulation modeling approaches have been proposed as one solution for reduced simulation modeling effort. A generic or a template-based simulation modeling approach often consists of an available set of pre-built, ready to use, modeling objects, modules, or models of common simulation situations. Using these modules, an analyst would simply “switch on” or “switch off” the model parameters of the generic module to fit it to her system under study. Table 3 summarizes some efforts in the area of generic and template-based simulation modeling.

Table 3: Research works in the area of generic or template based simulation

Generic Simulation Models of Reusable Launch Vehicles (Steele et al. 2002)
The Generic-Specific Modeling Approach: An application of artificial intelligence to simulation (Mackulak and Cochran 1990)
Effective Simulation Model Reuse: A case study for AMHS modeling (Mackulak, Lawrence and Colvin 1998)
Simulation in a Box: (A Generic Reusable Maintenance Model) (Brown and Powers 2000)
Automatic Generation of Simulation Models from Neutral Libraries: An Example (Son, Jones and Wysk 2000)
Organization and Selection of Reconfigurable Models (Diaz-Calderon, Paredis and Khosla 2000)
Composable Simulations (Kasputis and Ng 2000)
Observation on the Complexity of the Composable Simulation (Page and Opper 1999)

Ozdemirel and Mackulak (1993) found that although there are advantages and disadvantages associated with the type of approaches taken towards generic simulation model development, most suffer from efficiency problems. According to them, an ideal environment should assist a simulation analyst. This assistance may include model abstraction, data analysis, model generation, experimental design, and output analysis.

Steele et al. (2002) classify the area of generic simulation into two methodologies: (a) developing models applicable to more than one system; (b) developing a library of modules which assist in composing the simulation models. The authors propose a methodology for development of a systems-level generic model. It is suggested that developing a generic simulation modeling tool that assists an analyst in defining the conceptual model is a more robust approach that will have a larger user base and reduced chance of becoming obsolete. This is due to the fact that such a tool captures and encapsulates the information regarding the system components and their input parameters rather than providing executable components that are simulation programming platform specific.

2.2 Methodology and System Architecture

This research does not intend to implement “software/programming-level reusable simulation components.” Rather the work is intended to develop a framework that will assist a simulation analyst in the conceptual model development. Once the conceptual model is developed, the analysts may select the simulation software or programming platform of their choice. Specifically, the research will provide a framework that assists an analyst in identifying the significant input modeling parameters important in modeling a physical security system. Key aspects of the framework include:

- Identifying and defining the data primitives and their input parameters,
- Identifying and building the logical assemblies of the system components, and
- Building the common templates that define the relationships among the various system components.

A high-level view of the methodology for developing the modeling infrastructure is illustrated in Figure 1. The first task was to identify the categories that will hold the simulation primitives. There are three categories: object/entity, model and experimental. The identification and definitions of these categories were influenced by their counterparts in the SIMAN simulation modeling language. The object or the entity category defines the primitive that represent the work objects that request service from a system. These could be a person (such as a passenger at an airport), a non-physical object (such as a wireless message passed between security personnel guarding a museum) or a physical object (such as a piece of check-in luggage belonging to an airline passenger). The model category consists of primitives that represent their real world counterparts and perform any of the following actions:

- Create an entity
- Provide a waiting place for an entity
- Provide service to an entity
- Remove an entity

The experimental category defines those primitives that provide guidelines for the logical processing that is required in a simulation model; or those that effect the processing in its referencing primitives.

The second task of this research was to identify and define the primitives (and their input parameters) for physical security systems and classify them into any of the three defined categories. The simulation modeling structure and components from simulation languages and software (*e.g.* SIMAN, ARENA, EXTEND, SIMUL8, PROMODEL) were studied. A total of 14 primitives with 117 parameters are identified and categorized; one in the *entity* category, four in the *model* category and nine in the *experimental* category. Table 4 shows an example of the entity primitive along with the system data parameters and their explanations. The table has four columns. The first column, *Parameter Name*, contains the name of the configurable parameter for the primitive being defined. The second column of the table defines the *Parameter Type*. Parameter type can have the following values:

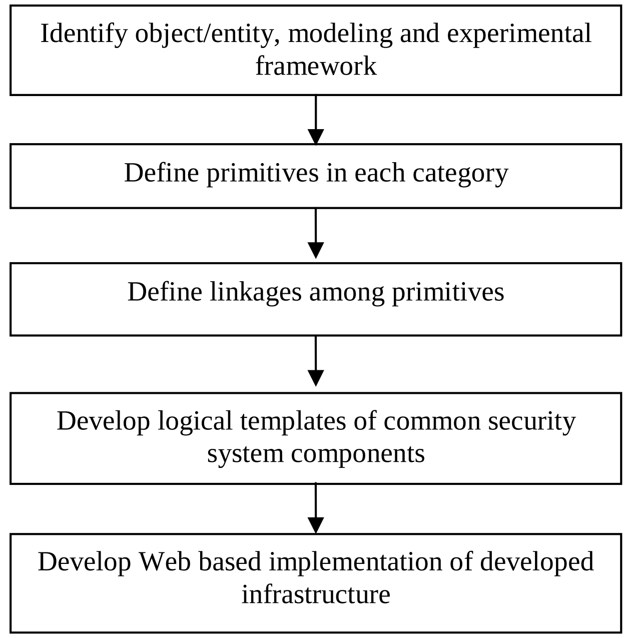

Figure 1**:** Research Methodology

- *Reference* - A reference parameter type means that the corresponding parameter is a reference to another simulation primitive type. For example the primitive *Work Center* has a parameter *Resources*. This parameter is of reference type since in the simulation model it will refer to a *Resource(s)* type model primitive
- *Native* – A native parameter type means that the corresponding parameter is native to the defined simulation primitive. For example, the primitive type *Entity* has a parameter *Length*. This parameter is of native type because it is a *distribution type* value defining the physical length of the defined primitive's instance

The third column, *Value Types*, lists the types of value that can be assigned to the parameter. The fourth column, *Description*, provides an explanation of the parameter.

The third task in this research involved defining the linkages and relationships among the identified primitives. The methodology for defining the associations and relationships is based on the principles of object-oriented systems analysis and design. After defining the associations and relationships, logical templates for common physical security system implementations were built. These templates were formed by grouping and relating the simulation primitives to represent real world sub-systems so that they promote component-based simulation modeling. The templates were developed by narrowing down the operations in physical security systems into smaller modules and mapping the real system components into flexible and modifiable conceptual simulation templates. Information about the security system equipment was collected and simulation modeling relevance data for these was extracted. Additional modeling relevance data from other sources, such as previous simulation studies, modeling primitives, what is used in simulation programming languages and software, was collected. A total of 15 templates were identified and defined. The developed templates embody the information that is relevant for performing the simulation when the object/equipment is part of a bigger system or needs to be individually modeled. For each identified template, a configuration table that defines its architecture (component primitives) is defined. Additional tables displaying the configuration of the component simulation primitives are also defined. These tables are reduced forms of the simulation primitive configuration tables defined during the second research task.

All the identified and developed templates are classified into five security sub-system categories: (1) Inspection and Detection System, (2) Identity Management System, (3) Perimeter Protection and Intrusion Detection System, (4) Access Control System, and (5) Entity Handling System. A single template may fall under one or more subsystems. Table 5 depicts this classification.

3 EXAMPLE

In this section, an example of the developed infrastructure is depicted. The infrastructure is applied to a scenario in which a simulation study is to be performed for estimating operational parameters (*e.g.*, % busy time, % idle time of operator(s) and equipment.) of an Explosive Detection System (EDS). An EDS is installed at an airport for screening of passenger check-in luggage. This example will highlight output of the framework that would be generated by the web-based implementation of the developed infrastructure.

Table 4: Configuration parameters of an entity simulation primitive

Parameter Name	**Parameter Type**	**Value Types**	**Description**
Name	Native	String	Unique name of the entity. The created simulation type may be referred by the string value of this parameter
Width	Native	Distribution	Physical width of the entity. This is used when the entity is being transported using a conveyor, passing through a work center, traveling on a path or when batched/grouped in the simulation model
Length	Native	Distribution	Physical length of the entity. This is used when the entity is being transported using a conveyor, passing through a work center, traveling on a path or when batched/grouped in the simulation model
Height	Native	Distribution	Physical height of the entity. This is used when the entity is being transported using a conveyor, passing through a work center, traveling on a path or when batched/grouped in the simulation model
Weight	Native	Distribution	Weight of the entity. This dimension is used when the entity is being transported using a conveyor, passing through a work center, traveling on a path or when batched/grouped in the simulation model
Priority	Native	Distribution	Processing priority level of the entity. Used when there are priorities that need to be given when selecting among a group of entities
Speed	Native	Distribution	Speed with which the entity moves freely in between work centers in the simulation model. This speed may be reduced due to 'jams' in the simulation model. It may also be increased when the entity is being transported in the model

Table 5: Classifications of level-one objects into security sub-systems

Inspection and Detection System	**Identity Management System**	**Perimeter Protection and Intrusion Detection System**	**Access Control System**	**Entity Handling System**
Explosive Detection Machine (including X-ray Inspection, Mail Room X-ray Inspection Machine)	Automatic Vehicle Identification (AVI) Machine	Communications Transceivers	Automatic Vehicle Identification (AVI) Machine	Laser Measurement Equipment
Handheld Metal Detector	Biometric or Touchpad Access Control Device	Entrance Door (Slide, Swing and Rotation and Turnstiles)	Biometric or Touchpad Access Control Device	
K-9 Unit	Card/Ticket Reader Machine		Card/Ticket Reader Machine	
Mail Purification Equipment	License Plate Recognition (LPR) Machine		Entrance Door (Slide, Swing and Rotation and Turnstiles)	
Mobile X-ray Inspection Machine	Token Dispenser Machine			
Walk-through Metal Detector				

Consider the high level function view of the EDS system as depicted in Figure 2. Since the developed infrastructure has a built in EDS template that is comprised of the primitive elements (shown inside the gray background in Figure 2). Using the web-application, an analyst would select the EDS template to be included in the conceptual model. The other primitives would be selected from the *primitives list* of the developed infrastructure. Table 6 highlights key parts of the output composition of the final conceptual model. Table 7 shows one of the many tables of input parameter requirements for the model. The first column of the latter tables shows the name of the simulation primitive parameter and the second column displays whether the parameter is required or not. In the second column a value of *Yes* means that the primitive parameter is required, *Optional* means that the primitive parameter is optional and its requirement depends upon the simulation study under consideration. A value of *No* means that the primitive parameter is not required in an instance of the template. It is assumed that the analyst would provide the names/values shown in *Value* column when prompted by the systems during creation of the conceptual model. For this example, artificial data has been inputted.

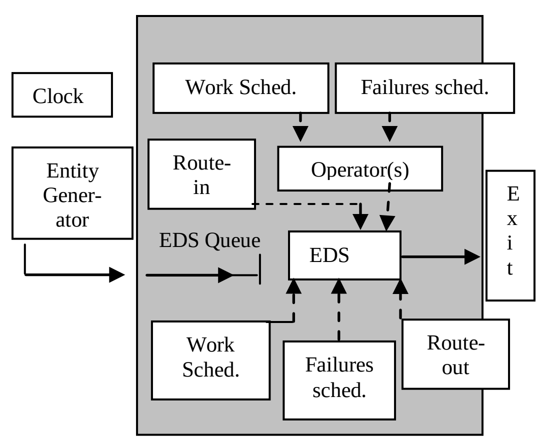

Figure 2: EDS System overview

Table 5: Conceptual model of EDS

Primitive Type	Category	Explanation
Entity	Entity	Representing real world luggage
Entry Point	Model	Creates the luggage entity
Queue	Model	Waiting place for the luggage arriving at the EDS
Work Center	Model	Simulation representative of real world EDS
Exit Point	Model	Object to remove the entities from the model
Clock	Experimental	Object to configure the simulation run parameters
Route-in	Experimental	Performs the function of providing selection rules from EDS workstation queue
Route-out	Experimental	Performs the function of directing the entity from the EDS workstation
Resources	Experimental	Simulation representative of real world operator(s) for the EDS
Work schedule	Experimental	One each to configure the work schedule of the EDS and operator
Failures schedule	Experimental	One each to configure the failures schedule of the EDS

Table 6: Configuration parameters of the EDS work center

Parameter	Requirement	Value	Explanation/Assumptions
Name	Yes	*EDS*	Unique name of the work center.
Number of	Yes	*1*	Consider there is a single EDS
Resource(s)	Yes	*(EDSOperator)*	An array containing reference to resource(s) associated with this work center. In the current scenario it is a single cell containing the reference to the single EDS Operator resource
Resource Requirements	Yes	*Yes*	Guideline that define if the resource(s) is required before accepting work item(s)
Resource Release Guidelines	Yes	*1*	If resource(s) should be present always or could it be released for other possible work center. This may be defined by fraction of processing time defined for this work center. In the defined scenario this means that the resource must be present throughout the scanning operation
Setup Time	Yes	*NORM(2,1.3)*	Statistical distribution that defines the loading time at the EDS. Assume the value used
Processing Time	Yes	*NORM(3,1.2)*	Statistical distribution that defines the duration of the process or time delay when scanning is performed at the EDS. Assume the value used
Release time	Yes	*NORM(1,1.2)*	This is the amount of time that is spent to unload the entities after the scanning is performed. Assume the value used
Splitting	Yes	*No*	Defines if the arriving entity is a batched entity and it need to be split
Work Schedule	Yes	*EDSWorkSchedule*	Work schedule associated with this EDS. This parameter also defines the capacity of the EDS (i.e., number of entities that this work center can process simultaneously)
Failures Schedule	Yes	*EDSFailSchedule*	Failure schedule associated with this EDS

4 SUMMARY

This paper explains a template-based framework for assisting a simulation analyst in creating the conceptual model of a physical security system. The key significance of this framework is that it:

1. focuses on identifying variables and components/parameters that need to be collected,
2. allows for a top-down or bottom-up approach to develop the conceptual model,
3. encourages model reusability,
4. is implementation language independent,
5. provides the conceptual framework that supports an object-oriented model design,
6. enhances development of more modular and reusable components,
7. and is a maintainable and expandable architecture,

By assisting an analyst in defining the components and parameters of the conceptual model, the success of the simulation is more dependent on a defined scientific methodology rather than simply the skill of the analyst.

REFERENCES

Annino J. S., and E. C. Russell. 1979. The Ten Most Frequent Causes of Simulation Analysis Failure and How to Avoid Them. *Simulation* v 60: 137-140.

Brown, N., and S. Powers. 2000. Simulation in a Box: (A Generic Reusable Maintenance Model). In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1050-1056.

Diaz-Calderon, A., C. Paredis, and P. Khosla. 2000. Organization and Selection of Reconfigurable Models. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 386-393.

Gatersleben, M. R., and S. W Weij. 1999. Analysis and Simulation of Passenger Flows in an Airport Terminal. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1226-1231.

Jordan, S. E., M. K. Snell, M. M. Madsen, J. S. Smith, and B. A. Peters. 1998. Discrete-Event Simulation for the Design and Evaluation of Physical Protection Systems. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 899-904.

Joustra, P. E., and N. M. V. Dijk. 2001. Simulation of Check-In at Airports. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1023-1028.

Kasputis, S., and H. C. Ng. 2000. Composable Simulations. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1577-1584.

Keller L, C. Harrell, and J. Leavy. 1991. The Three Reasons Why Simulation Fails. *Industrial Engineering* 23 (4): 27-31.

Kyle, R. G. 1998. Washington Dulles International Airport Passenger Conveyance Study. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 1119-1123.

Leone, K. 2002. Security System Throughput Modeling. Transportation Security Administration, Aviation Security Research and Development Division, Atlantic City, New Jersey.

Mackulak, G. T., F. P. Lawrence, and T. Colvin. 1998. Effective Simulation Model Reuse: A case study for AMHS modeling. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 979-984.

Mackulak, G. T., and J. K. Cochran. 1990. The Generic Specific Modeling Approach: An Application of Artificial Intelligence to Simulation. In *IIE Integrated Systems Conference & Society of Integrated Manufacturing Conference Proceedings*, 82-87.

Ozdemirel, N. E., and G. T. Mackulak. 1993. A Generic Simulation Module Architecture Based on Clustering Group Technology Model Codings. *Simulation* 60:6, 421-433.

Page, E. H., and J. M Opper. 1999. Observation on the Complexity of the Composable Simulation. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 553-560.

Parizi, M. S., and J. P. Braaksma. 1995. An optimum Resource Utilization Plan for Airport Passenger Terminal Building. *Transportation Research Record* TRB, National Research Council, Washington, D. C. 1506: 34-43.

Robinson S. 1999. Three Sources of Simulation Inaccuracy (And How to Overcome Them). In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1701-1708.

Robinson S., and M. Pidd. 1998. Provider and Customer Expectations of Successful Simulation Projects. *Journal of Operational Research Society* 49: 200-209.

Rowe, J. 1960. Towards a Theory of Scheduling. *Journal of Industrial Engineering*. Vol. 11, no. 2: 125-136.

Saffarzadeh M., and J. P. Braaksma. 2000. Optimum Design and Operation of Airport Passenger Terminal Building. *Transportation Research Record* TRB, National Research Council, Washington, D. C. 1703: 72-82.

Smith, J. S., B. A. Peters, S. E. Jordan, and M. K. Snell. 1999. Distributed Real-Time Simulation for Intruder Detection System Analysis. In *Proceedings of the 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 1168-1173.

Son, Y. J., A. T. Jones, and R. A. Wysk. 2000. Automatic Generation of Simulation Models from Neutral Libraries: An Example. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1558-1567.

Steele,
M. J., M. M. G. Rabadi, and G. Cates. 2002. Generic Simulation Models of Reusable Launch Vehicles. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C. H. Chen, J. L. Snowdown, and J. M. Charnes, 747-753.

Willemain T. R. 1995. Model Formulation: What Experts Think About and When. *Operations Research* 43 (6): 916-932.

AUTHOR BIOGRAPHIES

ASHU GURU is a Doctoral Student with the Department of Industrial and Management Systems Engineering at the University of Nebraska. He earned his Bachelor's degree in Mechanical Engineering from Delhi College of Engineering, (Delhi, India) and his M.S. in Manufacturing Systems from the Industrial Engineering department at the University of Nebraska. His research interests include information systems, database management systems, and discrete-event simulation <ashuguru@yahoo.com>.

PAUL SAVORY is an Associate Professor in the Department of Industrial and Management Systems Engineering at the University of Nebraska. Having degrees in industrial engineering, operations research, and computer science, he has taught numerous courses and consulted in the areas of computer simulation, operations research, and statistics. His email address is <psavory@unl.edu>.