



NORTH-HOLLAND

Experimental Evaluation of a Simulation Environment for Information Systems Design

James R. Warren

School of Computer and Information Science, University of South Australia, The Levels, SA 5095, Australia

A. F. Norcio, Jack W. Stott, and G. C. Canfield

Department of Information Systems, University of Maryland, Baltimore County, Baltimore, Maryland

This article presents an experiment assessing the decision support value of a simulation environment for the information systems (IS) design process. We have implemented a prototype simulation environment that uses data flow diagrams (DFDs) augmented with the performance rates of system components to specify the structure and dynamics of IS designs. The DFD-based representation is automatically mapped to a stochastic queuing network simulation. Knowledge-based help supports formulation of simulation run parameters and interpretation of output. We measure the prototype's impact on system dynamics assessment via the accuracy of IS professionals' responses to questions about the dynamics of four IS cases. The prototype's simulation capability has a significant positive effect on accuracy scores for questions involving waiting times, system times, and queue lengths of jobs or customers. Subjects choosing to conduct more and longer simulation runs provide significantly more accurate assessments than less-active users of simulation. The findings suggest that IS professionals can make use of simulation technology within a compact time frame if the proper supports are provided; on-line help or other methods should be used to encourage users to conduct sufficiently long simulation runs; and embedding the prototype's capabilities in a computer-aided software engineering workbench would result in better designed information systems.

Address correspondence to Dr. Anthony F. Norcio, Department of Information Systems, University of Maryland, Baltimore County, 5401 Wilkens Avenue, Baltimore, MD 21228-5398. e-mail: norcio@ifsm.umbc.edu

1. INTRODUCTION

Information systems (IS) developers frequently benchmark the performance of the hardware and software components of proposed IS designs; but modeling the overall dynamics of an IS design is not normally part of the system development process. For the purposes of this research, we consider the analysis of overall IS design dynamics to entail modeling the extent to which a set of system components in a design (computer-based systems, people, and non-information-processing machinery interacting with the IS) satisfies its performance requirements. Dynamic modeling will assess such factors as the amount of time customers or jobs spend waiting for service, system throughput, and utilization of system resources. We believe that it is unfortunate that the analysis of IS dynamics is neglected, because performance factors are important considerations in many design decisions. The goal of our research is to make analysis of overall IS design dynamics a routine part of IS development.

Tools with proven capability to model IS dynamics have been available for decades; the simulation 4GL GPSS, for instance, was introduced in the early 1960s and has been broadly and successfully applied (Schriber, 1991). So why is the use of simulation modeling of IS designs not a common practice? A number of factors could be proposed:

- IS developers are unaware of simulation technology.

- The cost in computer run time of conducting a dynamic analysis is too high.
- The tools are too expensive.
- Dynamics are not a major issue in many projects, such as report generation.

While each of these points has a grain of truth, we believe there are ample projects in which none of these is the key factor. We see the key inhibitor to the use of dynamic modeling in IS design to be the issue of *usability*. Dynamic models must be made more usable, more convenient to IS analysts.

Usability of a dynamic modeling technique can be divided into the problem of specifying the model and the problem of using the model to answer questions. To partially address the first of these usability challenges, we have proposed an approach of integrating simulation modeling with the developers' computer-aided software engineering (CASE) platform. The *CASE / simulation system* designates an architecture allowing the automatic production of system simulations of IS designs from data in a CASE tool repository without the writing of computer simulation programs by the analyst (Warren, 1992). CASE/simulation makes performance modeling more convenient because the model is specified based on the existing system design notations used in the CASE platform. However, the issue of actually using the simulation model to answer questions remains. In the case of *stochastic* (i.e., random number-based) simulation, the analyst must understand

- the issue of getting a sufficient sample size—conducting a long enough simulation run and/or a sufficient number of replications of the simulation
- the correct interpretation of the simulation output, namely, as confidence intervals rather than exact values.

Park and Mellichamp (1990) demonstrate an expert system for simulation modeling knowledge. This provides a precedent for on-line help to support analysts in the use of stochastic simulation for answering questions about IS designs.

This article describes an experiment designed to determine the impact of CASE/simulation on the accuracy of IS professionals' assessment of the performance of IS designs. These are the latest findings in a series of investigations concerning the effect of simulation models in the IS design process. Description of our prototype CASE/simulation architecture and its application can be found in Warren and Stott (1992) and Warren et al. (1992b, 1992c). The pilot

study for this experiment is described in Warren et al. (1992a). This study is described in full detail in Warren (1992). Our specific objectives in this study are to investigate the hypothesis that CASE/simulation systems lead to more accurate assessment of IS design dynamics and suggest further prototype developments and experiments that will lead to an increased understanding of decision support methods in this domain.

The next section reviews literature influencing the design and motivating the development of a prototype CASE/simulation system and lists the research questions about CASE/simulation. The third section describes the experimental method used to address key research questions. The fourth and fifth sections report experimental results and draw conclusions from the results, respectively.

2. BACKGROUND

The literature suggests a growing interest in dynamic modeling of IS designs. Boydston et al. (1980) offered an early effort at integration of simulation with system specification. They focused on production of SIMSCRIPT simulations from system descriptions in a variant of the Problem Statement Language (PSL; Teichrow and Hershey, 1977). More recent efforts are reflected in Cadre's Teamwork/SIM module, which analyzes specifications from their CASE tool and reveals design errors and processing bottlenecks (Pallatto, 1990). Many other modeling approaches focus on a specific component of the overall IS. For instance:

- *Database*. Eich et al. (1989) presented a methodology for the simulation of database architectures for performance evaluation. This approach allows an analyst to simulate the benchmark performance of a given database architecture in a specified hardware/software environment.
- *Software*. A prototype system has been developed to support the performance analysis of high-performance software as an integral part of the software development process (Ammar, 1991). This tool allows developers to consider the performance impact of implementational considerations (such as whether to use arrays or linked lists) in a machine-independent manner before the writing of code.
- *Networks*. The Network Simulation Testbed (NEST; Schwartz et al., 1990) is a graphical environment for simulation and rapid prototyping of distributed networks and network protocols. Users develop simulations of communications networks

using a set of graphical tools. Analysts can perform functions on nodes and links even while the simulation is running to simulate transient behaviors (such as a node going down or addition of a new node) that are typically difficult to study with analytic methods.

And there have been efforts at organizational (and even interorganizational) dynamic modeling tools. Dur and Bots (1992) present a graphical environment for dynamic modeling of organizations based on task/actor simulation. Their environment includes editors for tasks and actors, as well as a custom simulation engine. Jordan and Evans (1992) use a custom simulation language, SIMIAN, to simulate IS strategy.

There is also a growing interest in techniques that make simulation more usable. Many of the recent tools are in fact "shells" or environments around the harder-to-use, traditional simulation tools. The SASOS system (Humphreys et al., 1992) uses the simulation capabilities of Design/CPN¹ for organizational modeling, but a custom application developed in Apple Computer's Hypercard 2 is used for its business information repository, the SASOS input specification. Streng and Sol (1992) present an approach (i.e., something less tightly integrated than a "tool," per se) in which interorganizational dynamics are represented in terms of layered actors, networks, and entities (LANE); the LANE representation can then be simulated (and animated) using SMC's Siman/Cinema. There is a general trend to use graphics, dialog boxes, and animation to frame the model so that it is more easily specified and comprehended; commercial examples include QASE for network configuration (Gore, 1990) and CACI's Sim-Process for business processes. Also, there is the issue of providing on-line help to the analyst/user. Frankel and Balci (1989) describe the help system for the simulation model development environment (SMDE; Balci and Nance, 1987). After reviewing the characteristics of "good" on-line assistance, they produce a help system with a general assistance manager and "local" tool-specific help.

We have developed a prototype CASE/simulation system to embody what we believe are the most promising practices with respect to usability of dynamic models for IS design decision support. The prototype automatically produces stochastic discrete event-based simulation models from a repository of

data flow diagrams (DFDs) augmented with information describing the performance of system components.² Each DFD process is viewed as a server/queue in a queuing network. The prototype provides simulation in the context of an integrated simulation environment under the X Window System³ graphical user interface. Knowledge-based, intelligent help is provided in the formulation of simulation run parameters and in interpretation of stochastic simulation output. Help features include

- on-line definitions of output statistics
- confidence intervals generated automatically for the simulation output statistics
- summaries of process performance, including heuristic detection of bottlenecks
- computation of necessary warm-up periods
- on-line explanation of the impact of run duration and number of replications of confidence intervals.

Figure 1 illustrates the major features of the prototype. A more complete description of the environment can be found in Warren et al. (1992c).

The prototype simulation environment was designed to function as the apparatus for usability experimentation. This design goal, along with the objective of supporting IS analysts (rather than some generic user), led to a number of interesting design decisions:

- The system uses the method of replications to derive confidence intervals. This is not the most computationally efficient method, but we believe it to be the most easily understood. See Pawlikowski (1990) for a survey of stochastic simulation methods.
- The prototype is oriented toward the analysis of mean performance, as distinct from the emphasis on worst-case performance common in the analysis of "hard" real-time systems (Stonyenko et al., 1991).
- Output statistics related to utilization, throughput, and queuing are given in a tabular form for each DFD process. Graphics and animation were not

¹ Design/CPN is a proprietary product of MetaSoftware Corporation.

² DFDs were selected because they are graphical, supported by current CASE technology, and familiar to most IS analysts. Additionally, there is precedent for the use of semantically extended DFDs for advanced systems analysis approaches (Babin et al., 1991; France, 1992) and for dynamic systems in particular (Ward, 1986).

³ X Window System and X11 are trademarks of the Massachusetts Institute of Technology.

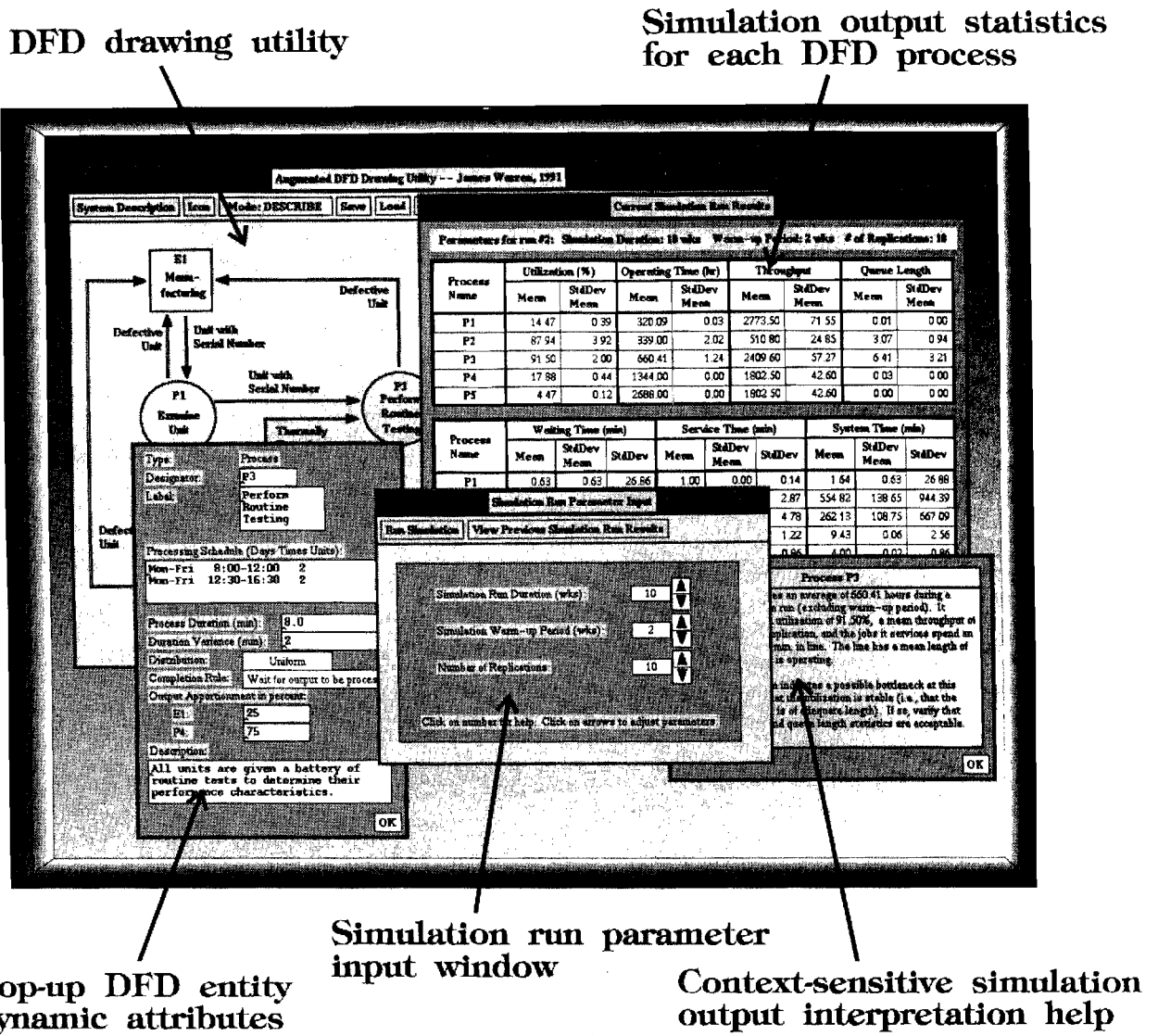


Figure 1. The prototype IS design dynamics simulator runs under X11 on an HP9000 workstation.

used. Determining their impact is left as a separate research problem.

- A custom DFD tool was used for the experiment. An earlier version of the prototype was interfaced to Excelerator's data dictionary (Warren and Stott, 1992). For the experimental context, we sought the tightest possible integration of the environment and wished to avoid any learning curve associated with a separate CASE tool.

The question of principal interest to us is this: Once the analysts have input a DFD-based model of the proposed system, does the prototype help them to answer useful, realistic questions about the performance of IS design cases? Thus, given a simula-

tion environment providing automatic simulation and intelligent help, the following research questions are posed:

1. Does the simulation environment allow IS analysts to evaluate more accurately the dynamic performance of IS designs than an environment providing DFDs augmented with dynamics attributes but lacking simulation capability?
2. Which, if any, features of this simulation environment are associated with more accurate evaluation of the dynamic performance of IS designs?

The remainder of this article describes experimentation addressing these two research questions.

3. METHOD

For purposes of this experiment, the prototype CASE/simulation system was developed so that its software environment can be invoked in one of two modes: 1) a mode that allows viewing of DFDs augmented with dynamics attributes but provides no simulation capability; or 2 a mode that, in addition to the DFD viewing of mode 1, provides queuing network-based simulation of IS designs, including expert help on simulation experiment designs and output analysis. Given these two modes, we define two experimental treatment levels.

Level A. Subjects answer questions about IS designs using the prototype in mode 1; they are provided with DFDs annotated with dynamics information, but have no simulation capability.

Level B. Subjects answer questions about IS designs using the prototype in mode 2; they are provided with DFDs annotated with dynamics information, queuing network-based simulation of the IS design represented in the DFD, and expert help on simulation experiment design and output analysis.

Subjects were given case problems to analyze and asked to answer a series of questions about the dynamics of IS designs. The subjects attempted to address various sets of questions at the two treatment levels. The observed mean difference in accuracy between treatment levels is expected to lead to the rejection of the null hypothesis that IS dynamics evaluation accuracy measures are equal for treatment levels A and B. The use of the software environment is also observed to support analysis of the association of aspects of tool use with accuracy.

A repeated-measures design is used in which each subject responds to each treatment level. Each subject analyzes and answers dynamics questions on four IS design cases, two cases at each level. A pilot version of this study has been conducted using IS graduate students as subjects (Warren et al., 1992a). Pilot study results are used as the basis for various assumptions in the method. The method (including all dynamics cases and questions and handouts to subjects) and results (including individual responses to questions) are described completely in Warren (1992).

3.1 Dependent Measures

The dependent measure for accuracy in assessment of IS dynamics is determined via subject responses to quantitative questions regarding the dynamics of

IS designs. Eight questions were asked regarding each of four cases.

The dynamics cases are evenly balanced between service and manufacturing. The first two cases are based on illustrated cases in an IS textbook by Saldarini (1989). These cases concern a videotape rental store and are representative of service-oriented systems. The second pair of cases concern a small high-tech manufacturing operation and are drawn from the authors' own experiences. All of the cases are somewhat modified from their original forms. In particular, the cases are simplified to make the assessment of their dynamics more feasible within the limiting time constraints given to the experimental subjects. One of the four cases, Video Tape Return, is given in Appendix A with its associated dynamics questions.

Eight dynamics questions are formulated for each case. Dynamics questions are chosen to relate to concerns likely to arise due to the performance requirements of the analysed system. For example, in the videotape return case, the questions focus on how long customers wait, how the rental clerk's time is spent, and how quickly inventory is returned to stock shelves. The responses to these questions characterize the system dynamics relevant to the adequacy of the design and the potential value of introducing further automation.

The questions can be separated into four categories along two dimensions (Table 1). These dimensions are based on the type of information required to answer the questions. A first dimension of dynamics question is *expected* versus *emergent*. Expected questions are concerned with issues of throughput, service times, utilizations, and operating times of processors, where processing dependencies are straightforward. Expected questions can be addressed using straightforward analysis of the expected values of parameters of the queuing systems. Emergent questions are concerned with features of systems performance related to queuing: waiting times and queue lengths of jobs and customers, and system times, especially threads of processing (i.e.,

Table 1. Four Types of Dynamics Questions Along Two Dimensions

	Single Queuing System	Multiple Queuing Systems
Simple analytic reasoning	Expected-simple	Expected-composite
Complex reasoning (consideration of queuing)	Emergent-simple	Emergent-composite

the amount of time needed for a job to traverse several connected processes). Thus, the answers to emergent-type questions are seen to “emerge” from congestion in the system; analytical reasoning to assess these quantities is much more involved than for expected-type questions. The second dimension is *simple* versus *composite*. A simple question requires the analysis of only a single server/queue system, whereas a composite question requires consideration of multiple server/queues. Each dynamics question can be categorized as *expected-composite*, *emergent-simple*, or *emergent-composite*. Two questions of each type were presented for each dynamic case.

Each response receives an evaluation accuracy measure between 0 (poorest) and 1 (perfect accuracy). Answers are scored based on the square of the factor of deviation from the true value. So, for instance, if a subject estimates 200 minutes (or 50 minutes) when the correct answer is 100 minutes, the subject has missed by a factor of two, and therefore gets an accuracy score of 0.25 (1 divided by 2²). The details of the accuracy scoring mechanism are explained in Appendix B. With eight questions asked regarding each of the four cases, the resulting overall accuracy score is between 0 and 8 for each subject for each case. Analyzing just the emergent-type questions yields an accuracy score between 0 and 4 for each case.

As subjects worked to answer dynamics question, usage variables were collected to characterize the manner in which the software system is used. These variables include

- the number of simulation runs conducted
- the total duration of the simulation runs (in simulated weeks)
- the total warm-up period duration (i.e., the amount of simulation data discarded to get the system into its “steady” state before measurements are taken)
- the total number of replications
- the number of times the simulation run parameter help is consulted
- the number of times the simulation output help is consulted
- the number of times the DFD viewer annotation is consulted.

Finally, after each case, subjects were asked to rate their confidence in the accuracy of their responses to the dynamics questions on a seven-point scale.

3.2 Subjects

Twenty-one IS professionals participated in this study. Participants were required to have at least 3 years of professional experience. The volunteers had a mean of 8.5 years of professional experience with information systems, were employed in various private companies as well as state and federal government, and had jobs spanning all phases of the system development life cycle (with systems analyst being the modal job title). Seventeen of the participants had at least a Bachelor’s degree, and seven had at least a Master’s degree.

3.3 Apparatus

The primary experimental apparatus is the prototype CASE/simulation system software and associated hardware. The experiment was conducted using six HP9000 series workstations with 16-inch color screen running X11. At each experimental level, subjects were provided with two windows: a DFD augmented with dynamics information describing the IS dynamics case, and a question-asking window querying the user about the dynamics of the IS design. Subjects were also provided with *xcalc* (a standard X11 utility that emulates a pocket calculator), paper, and pencil. These two windows provided the subjects with the IS dynamics cases. In treatment level B, subjects also received the “Simulation Run Parameter Input” window (Figure 1), which provided access to IS design simulation and expert help in simulation usage and interpretation.

3.4 Procedure

Participation in the study took the form of an all-day workshop in which subjects (about four per workshop) participated in a 2 1/2-hour introductory session and two 90-minute problem-solving sessions. During the introductory session, subjects developed a common baseline of knowledge of modeling and simulation concepts and worked with the prototype on an introductory IS case. The two problem-solving sessions were completed during the day of the introductory session.

In each problem-solving session, subjects were presented with a description of an organization and DFDs with corresponding data dictionary entries for two component systems of that organization (videotape rental in session 1 and manufacturing in session 2). Subjects were given up to 15 minutes to peruse these materials. After reading about the organization, subjects were presented with an IS design dynamics case set in the organization. Subjects spent

30 minutes analyzing the case with the tools available (randomly, treatment level A, no simulation, or B, simulation), to answer the dynamic questions on screen. The responses to the dynamics questions and the usage variables were recorded at the end of the session. Subjects were then given 5 minutes to fill out a "Problem & Suggestions" form. After a 5-minute break, subjects were presented with another case (set in the same organization), given 30 minutes to answer the dynamics questions, and 5 minutes to fill out another "Problems & Suggestions" form. At the end of each 90-minute problem-solving session, subjects were given the worked solutions to the cases they had been analyzing.

3.5 Experimental Design

We chose to use a within-subjects design in which all subjects participate in both treatment levels because it is a more powerful experimental method than a between-subjects design (in which a subject experiences treatment level A or treatment level B, but not both). In a within-subjects design, if there is a high level of variation among the abilities of individual subjects, the variation is seen in both treatment levels and has little impact on the accuracy of the experiment. For a between-subjects design, many subjects are needed to "average out" this natural variability. Since getting the all-day participation of skilled IS professionals can be difficult, we found the within-subjects design to be attractive. Furthermore, since subjects get to participate in both treatment levels, all the IS professionals had the opportunity to use the simulator; this was an important selling point in attracting participants.

Within-subjects designs, however, introduce some difficulties. Learning effects over time are to be expected, so the experimental design must address the order of presentation of the treatment levels. That is, subjects may reasonably be expected to get better at working the cases as time goes on. Thus, if, for instance, subjects always perform the first case without simulation and then the second case with simulation, we would have no way to separate the effect of simulation from the subjects simply "getting the hang of it" and doing better on the second session than the first.

Table 2 shows the layout of a design that accounts for the carryover⁴ effects of one treatment on the

Table 2. Treatment Levels by Time for the Study

Ordering Group	Time			
	Problem-Solving Session 1		Problem-Solving Session 2	
	t_1	t_2	t_3	t_4
Group 0	A	B	A	B
Group 1	B	A	B	A

next (Stevens, 1986). Subjects are randomly assigned to one of two ordering groups. At time t_1 , subjects randomly receive either case 1 or case 2 to analyze using either the treatment A environment (if they are in ordering group 0) or the treatment B environment (if they are in ordering group 1). At time t_2 (the second part of the first problem-solving session), subjects receive either case 1 or 2, whichever they have not yet analyzed, using the treatment level A environment (if they are in ordering group 1) or B (if they are in ordering group 0). In problem-solving session 2 (times t_3 and t_4), subjects analyze cases 3 and 4 in random order using treatment level A and B environments in the order determined by their ordering group membership. In this design, the treatment effect is measured as the *interaction* of the ordering group membership and time. The important aspect of measuring the interaction is that we recognize that there may be a number of factors leading to variation in scores—learning, ordering, difficulty of cases, and so on—but we are measuring the average effect of simulation across the other variations. (See Appendix C for a more detailed discussion of the underlying statistical issues in this study.)

4. RESULTS

4.1 Treatment Effect on Response Accuracy

Figure 2 shows the overall accuracy scores (i.e., accuracy scores summed over all eight questions of a case) for the subjects. Table 3 shows the means and standard deviations of overall scores at t_1 - t_4 and for the sum of the scores at t_1 and t_3 (beginning of session) and t_2 and t_4 (end of session) by ordering group. It can be seen that the overall trend is as predicted: subjects score higher at treatment level B (t_2 and t_4 for ordering group 0 and t_1 and t_3 for ordering group 1). The treatment effect is significant at the 0.64% level (see Appendix C for a discussion of the statistical model and its assumptions). The observed treatment effect size on overall accuracy scores is 1.10 on eight-point accuracy scale. Correcting a factor of two estimation error on one question

⁴ *Carryover* is a general term for the effects of prior sessions on subsequent ones; typical carryover effects include learning and fatigue.

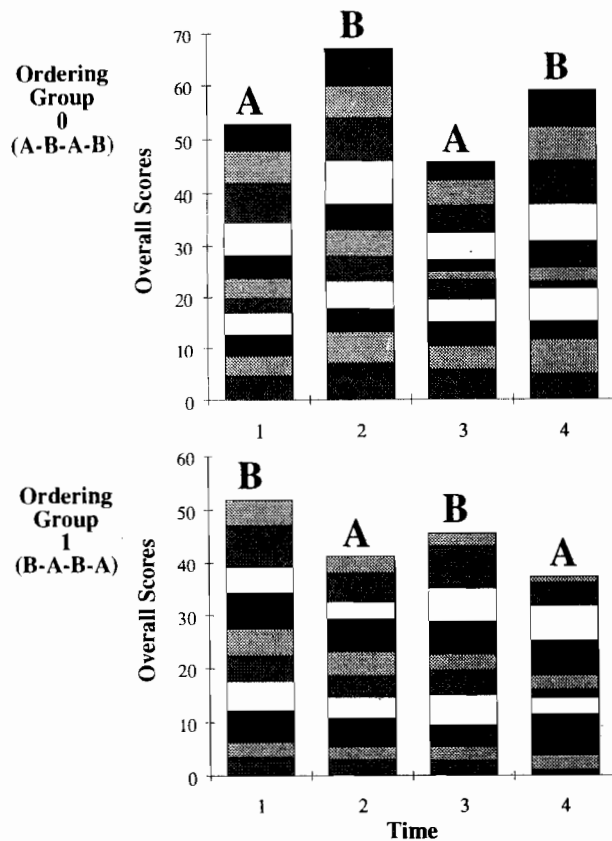


Figure 2. Overall accuracy scores for the 11 subjects in ordering group 0 and 10 subjects in ordering group 1. Individual scores are indicated by shading. Treatment levels are labeled above bars. Treatment effect is significant at the 0.64% level.

per case would result in a 0.75-point increase in the accuracy score (see Section 3.1 for explanation). Thus, some quantities are estimated substantially better with the availability of simulation.

Figure 3 shows the emergent-type accuracy scores (i.e., accuracy scores summed over questions 5-8 of a case) for the subjects. Table 4 shows the means and standard deviations of emergent scores at t_1 - t_4 and for the sum of the scores for the beginning session and end of the session. It can be seen that the trend is the same as with the overall scores: subjects score higher at treatment level B. The effect is much sharper for emergent-type questions than for the overall scores and is significant at the 0.04% level.

Table 5 shows the mean accuracy scores observed for each of the four question types (expected-simple, expected-composite, emergent-simple, and emergent-composite) by ordering group and time. There is no evident trend related to treatment level for the

Table 3. Observed Means and Standard Deviations (SD) of Overall Accuracy Scores by Ordering Group and Time

Ordering Group	Time	Treatment Level	Mean	SD
0	t_1	A	4.81	1.30
	t_2	B	6.11	1.26
	t_3	A	4.16	1.26
	t_4	B	5.35	2.06
	$t_1 + t_3$ (beginning of session)	A	8.96	2.16
	$t_2 + t_4$ (end of session)	B	11.46	3.11
1	t_1	B	5.19	1.49
	t_2	A	4.12	1.24
	t_3	B	4.56	1.88
	t_4	A	3.74	2.45
	$t_1 + t_3$ (beginning of session)	B	9.75	3.19
	$t_2 + t_4$ (end of session)	A	7.86	3.36

expected-type questions. In fact, the entire treatment effect on overall scores is due to the effect on emergent-type scores. There is no statistically significant treatment effect on the scores for expected-type questions. It is interesting that the significant treatment effect on emergent-type scores is well distributed between emergent-simple and emergent-complex types of questions at 0.52 and 0.61 points of accuracy improvement (on a 0-2 scale), respectively.

4.2 Analysis of Software System Usage

Performing linear regression of the total number of simulation runs conducted at treatment level B (NRUN) on accuracy scores yields significant regression lines. Regression on overall scores gives an estimated slope of 0.92 points with an intercept of 7.02, which is significant at the 0.36% level and explains 33.42% of the variance in overall scores. That is, mean accuracy scores go up by almost 1 point per simulation run conducted. Figure 4 plots the regression line with the observed overall scores. Regression on emergent scores gives a slope of 0.56 with an intercept of 2.92, which is significant at the 0.26% level and explains 35.56% of the variance in emergent accuracy scores. Figure 5 plots the regression line with the observed emergent-type scores. Subjects were encouraged to conduct an initial simulation run and then to consult the environment's recommendations for warm-up duration, run length, and confidence intervals. As such, conducting multi-

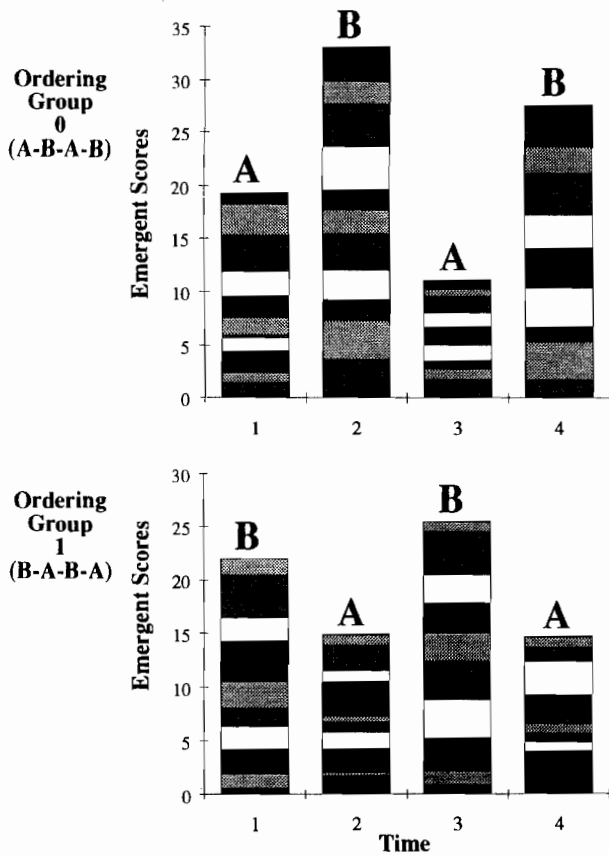


Figure 3. Accuracy scores for emergent-type questions for the 11 subjects in ordering group 0 and 10 subjects in ordering group 1. Individual scores are indicated by shading. Treatment levels are labeled above bars. The treatment effect is significant at the 0.04% level.

Table 4. Observed Means and Standard Deviations (SD) of Emergent-Type Accuracy Scores by Ordering Group and Time

Ordering Group	Time	Treatment Level	Mean	SD
0	t_1	A	1.75	0.92
	t_2	B	3.01	0.80
	t_3	A	1.01	0.49
	t_4	B	2.49	1.37
	$t_1 + t_3$ (beginning of session)	A	2.77	1.08
	$t_2 + t_4$ (end of session)	B	5.50	1.81
1	t_1	B	2.20	1.05
	t_2	A	1.49	0.90
	t_3	B	2.54	1.14
	t_4	A	1.47	1.28
	$t_1 + t_3$ (beginning of session)	B	4.74	2.01
	$t_2 + t_4$ (end of session)	A	2.96	1.90

ple simulation runs ($NRUN > 2$, or once per session) is expected to correlate with “good” use of the tool.

The total number of weeks simulated, SIMTOT, is highly correlated with the number of simulation runs, NRUN. SIMTOT is a more informative variable, however, because subsequent simulation runs tend to be longer than initial runs (to achieve narrower confidence intervals, for instance). Modeling emergent-type scores in session 2 as a linear function of SIMTOT explains 42% of the variance in session 2 emergent-type scores with a nominal significance of 0.14%. One might expect that a point of diminishing returns is reached after a certain duration has been simulated. A better fit may be achieved by using a less-than-linear relationship. Regressing on the cube root of SIMTOT produces a stronger model (as does regression on the log or square root of SIMTOT), explaining 62% of the variance in session 2 emergent-type scores with a nominal significance of 0.01%. While there is no reason to expect that there is anything special about the cube root transform, the overall finding is that there is a strong, positive, sublinear relationship between the intensity of simulation use by the subjects and their accuracy scores.

4.3 Survey Results

Responses to the multiple choice question on the “Problems & Suggestions” form regarding confidence in accuracy of responses are coded as integers in the range of 1 to 7. The confidence ratings data significantly violates normality, and analysis of correlations using Kendall’s τ -b [a nonparametric measure of association based on numbers of concordant and discordant pairs of observations (SAS, 1989)] is used in lieu of a more conventional analysis of variance. Table 6 shows correlation of confidence ratings with ordering group. The signs of all four coefficients favor the availability of simulation as having a positive association with subjects’ confidence (although only three of the four scores are significant at the 5% level). This provides some support for the notion that users “know what’s best” for themselves. They rate their confidence in their responses higher with the availability of simulation, which is, in fact, observed to improve the accuracy of their responses.

5. CONCLUSIONS

The prototype simulation environment is designed to “deliver” simulation technology to IS analysts. It

Table 5. Mean Scores by Time and Ordering Group, and Observed Treatment Effect Sizes per Case for Expected-Simple-, Emergent-Simple-, Expected-Composite-, and Emergent-Composite-Type Accuracy Scores

Score Type	Time	Ordering Group 0		Ordering Group 1		Observed Treatment Effect Size
		Treatment Level	Mean	Treatment Level	Mean	
Expected-simple	t_1	A	1.82	B	1.51	-0.10
	t_2	B	1.67	A	1.50	
	t_3	A	1.68	B	1.04	
	t_4	B	1.52	A	1.16	
Expected-composite	t_1	A	1.23	B	1.48	0.07
	t_2	B	1.44	A	1.14	
	t_3	A	1.47	B	0.98	
	t_4	B	1.34	A	1.11	
Emergent-simple	t_1	A	0.87	B	1.11	0.52
	t_2	B	1.49	A	0.55	
	t_3	A	0.89	B	1.37	
	t_4	B	1.40	A	0.97	
Emergent-composite	t_1	A	0.89	B	1.09	0.61
	t_2	B	1.52	A	0.94	
	t_3	A	0.12	B	1.17	
	t_4	B	1.09	A	0.50	

is known that a correctly implemented stochastic (random number based) simulator will produce results that converge toward the true mean performance values of the system it is modeling (Pawlikowski, 1990). Simulation technology has been available in the form of simulation languages for decades. The issue of concern to us is *usability*—embedding the stochastic simulation model in an environment such that IS analysts can readily use it to support design decision making. Toward this end,

the prototype environment incorporates features of simulation environments from various domains, expert systems in simulation, intelligent help systems, and CASE tools. We have experimentally assessed the environment on its ultimate criterion for success: Does it lead to better design decisions?

It is not self-evident that the availability of a simulation model automatically leads to correct understanding on the part of the analyst. The question of concern to the analyst will not always align per-

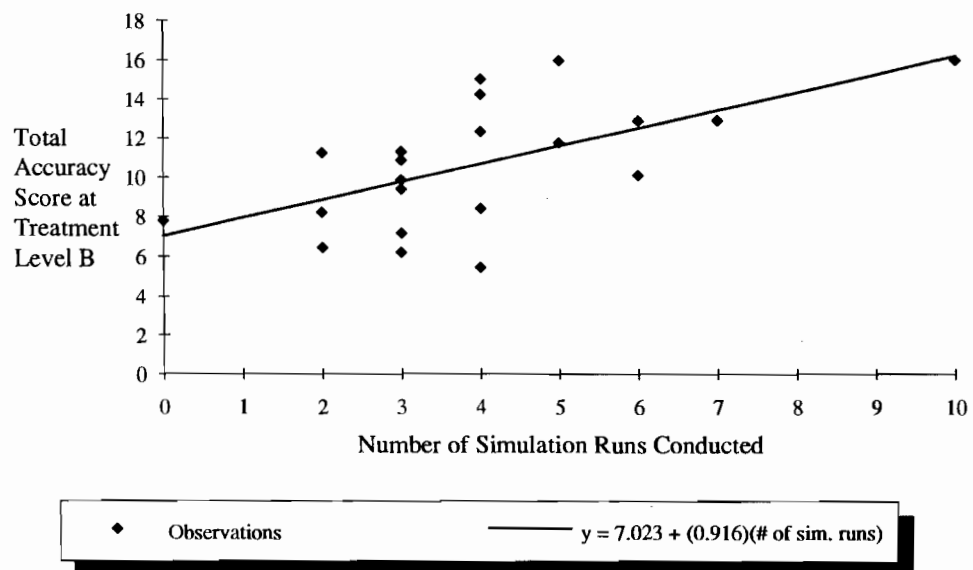
**Figure 4.** Linear regression of number of simulation runs conducted at treatment level B on overall accuracy scores.

Table 6. Correlation (Kendall's τ -b and its Two-Tailed Significance) of Subjects' Confidence Ratings with Ordering Group

Time	Treatment Level		Correlation coefficient	Significance %
	Group 0	Group 1		
t_1	A	B	0.437	3.66
t_2	B	A	-0.220	32.53
t_3	A	B	0.594	0.67
t_4	B	A	-0.455	4.47

fectly with the simulation output. The analyst must understand the definitions of the simulation statistics and apply them to understanding the system at hand. For instance, our prototype environment presents statistics per DFD process; but half of the dynamics questions asked in the study require consideration of multiple processes (that is the definition of the *complex* question category; see Section 3.1). In a simulator that allows definition of statistics to suit the analyst's current problem, the analyst must still possess an understanding of simulation output statistics to use the definition facility successfully. The simulation environment must support the analyst in interpreting the simulation model.

This study demonstrated an interesting area where simulation is observed *not* to be of benefit. For the expected-type questions (throughput, service times, utilizations, operating times), the subjects showed no difference in the accuracy of their responses whether the simulator was available or not. The simulator did

provide accurate estimates of quantities relevant to answering these questions. Nonetheless, it is evident that understanding the output was no easier for the subjects than working from the problem description to compute the answer with a calculator. Thus, for many types of quantitative analysis (those fitting the definition of expected-type questions), we observed no evidence that simulation is of value. This is not to say the simulation cannot be of value here. The answers observed are far from perfectly accurate; there is much room for improvement. It may be that animation would serve a useful role here, promoting in analysts the understanding of the system necessary to formulate the answers to computationally simple questions.

For emergent-type questions, availability of simulation shows a sizable positive impact on accuracy of responses. Emergent-type questions entail issues of queuing. It is not surprising that analysts would be unable to accurately estimate these quantities. Subjects were able to use the simulation output to significantly improve their responses versus their ability to estimate the answer via intuition and/or rough analytical approximations.⁵ Thus, in this area,

⁵ Since the emergent-type questions do not fit standard analytical queuing models, such as M/M/1, it is considered highly unlikely that any subjects applied exact analytical reasoning to these questions.

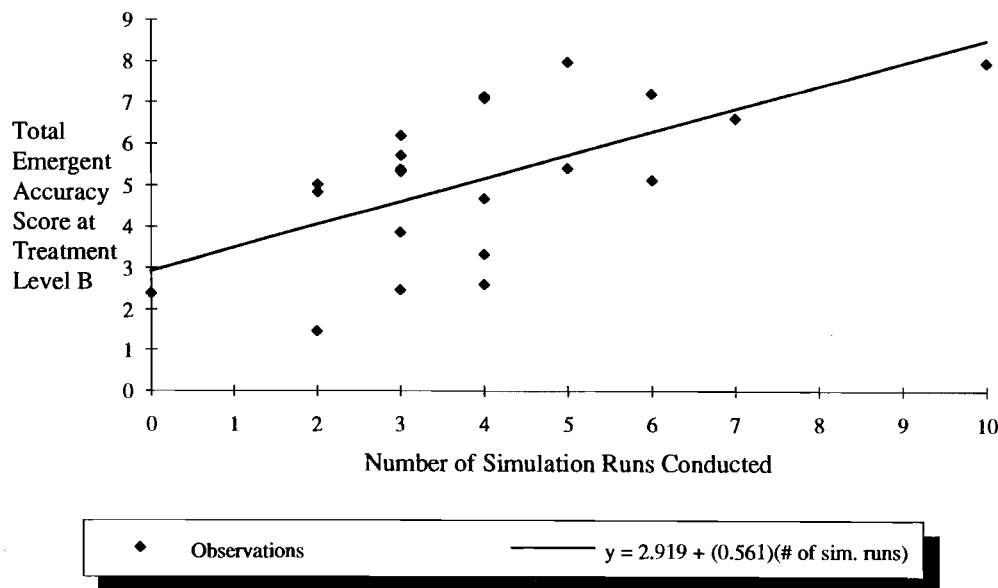


Figure 5. Linear regression of number of simulation runs conducted at treatment level B on emergent-type accuracy scores.

there is a successful delivery of simulation technology.

Simulation technology can be used successfully in a short time frame. The subjects participated in a 1-day workshop, beginning with 2 1/2 hours of initial training. The subjects were well-educated and experienced IS professionals. In addition to a short training period, the problem-solving sessions themselves were quite brief; after an initial orientation and reading period, subjects had 30 minutes to address eight questions. While this may seem a frantic pace, it should be noted that in this case the problem was already precisely framed: a DFD and data dictionary were provided, processing durations for the low-level tasks were given, and the questions were precisely framed. The analysis had time to review these materials. With this much up-front work already performed and a simulator at hand, we believe that the analysts were able to proceed directly to addressing the questions. If the analysts in such a simulation are unable to proceed, then the simulation environment is not making the model sufficiently easy to use.

Stochastic simulation environments should encourage users to conduct sufficiently long simulation runs. With stochastic (random number-based) simulation, estimates become increasingly accurate as the amount of simulation time increases. In this study, the concept of confidence intervals was explained during the training, and the prototype environment incorporated several help features to encourage attention to the accuracy of the simulation results (e.g., automatic confidence interval generation, on-line explanation of simulation run parameters). Despite these supports, there was a high variance in the total amount of simulation time used by the subjects. Not surprisingly, conducting longer simulations was observed to be highly correlated with accuracy of responses. The supports provided in the prototypes are passive—the user must ask the system for advice and is not compelled to use that advice. It may be that more active techniques such as pop-up warnings or restrictions on minimum number of replications may be appropriate. But it is difficult to see how a one-size-fits-all set of rules could verify that an analyst has sufficiently accurate simulation results to suit the specific decision needs at a given time (and to suit the amount of real time the analyst wishes to invest in simulation runs). We believe there is no substitute for understanding on the part of the user, and supports should be in place to support this understanding.

Embedding the prototype's capabilities in a computer-aided software engineering (CASE) work-

bench should result in better designed information systems. Simulation capability as delivered by the prototype leads to more accurate assessment of queuing-related aspects of the dynamics of IS designs. Queuing statistics are central to quantifying customer waiting times, manufacturing lead times, numbers of jobs in progress, the time for detection of errors and defects, the time for processing a request, and the time merchandise lies idle before being made available to customers—quantities critical to the acceptability of modern, nonbatch information systems. The observed impact of the prototype is on the order of correcting a factor-of-two error in one out of four such quantities.⁶

CASE/simulation accomplishes fundamental software engineering goals by moving forward in the system development life cycle (SDLC) the detection of errors in system performance assessment. Boehm (1976) showed (based on experience at TRW) that the cost of correcting a software error at the acceptance-testing phase of the SDLC is 10 times as high as correcting an error in the design phase. The goal of early error correction, as a means of achieving low-cost and reliable software development, is central to software engineering (and computer-aided software engineering). The proper use of simulation early in system design can prevent very profound sorts of errors that may be exceedingly difficult to correct later. Simulation allows the study of system dynamics in the design phase; otherwise it would be studied (at best) in the acceptance-testing phase via benchmarking and observation. With an accurate perception of system dynamics early in the SDLC, developers can decide which areas have critical performance concerns. Thus, in addition to avoiding the development of a system that is too slow (and the subsequent need for rework), an accurate dynamic model can help to avoid overengineering a solution where high speed is not necessary.

We do not mean to express any undue bias toward the value of simulation models in IS design compared with other techniques. Specifically, we model an IS as a network of queues and then evaluate that network via stochastic simulation. We favor this method as the queuing network model integrates well with the popular DFD notation, and stochastic simulation is a simple, general method to evaluate such networks, requiring few assumptions about in-

⁶ Recall from Section 3.1 that a factor-of-two error scores an accuracy rating 0.75 points lower than a perfectly accurate response. The observed effect of simulation is to improve accuracy by 1.13 points on the four emergent-type questions of each case.

put or service distributions. However, analytical evaluation of queuing networks also has its merits (in the potential for lower computational cost, for instance). There are numerous other models that have been applied to the dynamics of IS design, including colored Petri nets (Ang et al. 1992), transition rules (Nota and Pacini, 1992), and process algebra (Milne, 1993). Each of these methods represents a mature mathematical model that can provide valuable decision information to the design process. The great barrier is not the applicability of the method, but its usability. This usability can be enhanced by determining the cognitive barriers to successful use and providing appropriate supports. Empirical experiments are then the proper method to assess the success of these supports for human understanding.

ACKNOWLEDGMENT

Special thanks are owed to those who volunteered their valuable time to participate in this study, to Narcheel Nagaraj for his patient discussion of experimental design alternatives, and to the University of Maryland, Baltimore County, Academic Computing Services for their excellent support, including provision of facilities for the experiment.

REFERENCES

- Ammar, R., A Computer-Aided Design System to Develop High-Performance Software, *J. Syst. Software* 15, 139-147 (1991).
- Ang, J., Conrath, D., and Savolainen, V., Analyzing information systems using Petri nets: Operations-oriented methodology, in *Dynamic Modelling of Information Systems, II* (H. Sol and R. Crosslin, eds.), Elsevier Science Publishers, 1992.
- Babin, G., Lustman, F., and Shoal, P., Specification and Design of Transactions in Information Systems: A Formal Approach, *IEEE Transactions on Software Eng.* SE-17, 814-829 (1991).
- Balci, O., and Nance, R., Simulation Model Development Environments: A Research Prototype, *J. Operat. Res. Soc.* 38, 753-763 (1987).
- Boehm, B., Software Engineering, *IEEE Trans. Comput.* 1226-1241 (December, 1976).
- Boydston, D., Teichroew, Spewak, S., Yamamoto, Y., and Starner, G., Computer aided modeling of information systems, in *Proceedings, IEEE COMPSAC80*, 1980.
- Dur, R. C. J., and Bots, P. W. G., Dynamic modelling of organizations using task/actor simulation, in *Dynamic Modelling of Information Systems, II* (H. Sol and R. Crosslin, eds.), Elsevier Science Publishers, 1992.
- Eich, M., Fan, C., Sun, W., and Rafiqu, S., A Methodology for Simulation of Database Systems, *Simulation* 241-254 (June 1989).
- France, R. B., Semantically Extended Data Flow Diagrams: A Formal Specification Tool, *IEEE Trans. Software Eng.* SE-18, 329-346 (1992).
- Frankel, V., and Balci, O., An On-Line Assistance System for the Simulation Model Development Environment, *Int. J. Man-Machine Stud.* 31, 699-716 (1989).
- Gore, A., QASE to Configure Huge Systems, *Macweek* 20 (November 13, 1990).
- Humphreys, P., Berkeley, D., and Quek, F., Dynamic process modelling for organisational systems supported by SASOS, in *Proceedings of Third International Conference on Dynamic Modelling of Information Systems*, 1992, pp. 47-64.
- Jordan, E., and Evans, J. B., The simulation of IS strategy using SIMIAN, in *Dynamic Modelling of Information Systems, II* (H. Sol and R. Crosslin, eds.), Elsevier Science Publishers, 1992.
- Maxwell, S., and Delaney, H., *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, Wadsworth Publishing, 1990.
- Milne, G., *Formal Specification and Verification of Digital Systems*, McGraw-Hill, London, 1993.
- Nota, G., and Pacini, G., Querying of Executable Software Specifications, *IEEE Trans. Software Eng.* 18 (1992).
- Pallatto, J., Cadre to Expand Suite of CASE Workstation Tools, *PC Week* (November 26, 1990).
- Park, Y., and Mellichamp, J., A statistical expert system for simulation analysis, in *Proceedings, Summer Computer Simulation Conference*, 1990, pp. 611-616.
- Pawlikowski, Steady-State Simulation of Queuing Processes: A Survey of Problems and Solutions, *ACM Comp. Surv.* 22, 123-170 (1990).
- Saldarini, R., *Analysis and Design of Business Information Systems*, Macmillan, 1989.
- SAS Institute Inc., *SAS / STAT User's Guide, Version 6*, 4th ed. SAS Institute, Cary, North Carolina, 1989.
- Schriber, T., *An Introduction to Simulation Using GPSS / H*, Wiley, New York, 1991.
- Schwartz, A., Yemini, Y., and Bacon, D., NEST: A Network Prototyping Testbed, *Commun. ACM* 33, 63-74 (1990).
- Stevens, J., *Applied Multivariate Statistics for the Social Sciences*, Lawrence Erlbaum Associates, 1986.
- Stonyenko, A. D., Hamacher, V. C., and Holt, R. C., Analysing Hard-Real-Time Programs for Guaranteed Schedulability, *IEEE Trans. Software Eng.* SE-17, 737-753 (1991).
- Streng, R. J., and Sol, H. G., A dynamic modelling approach to analyze chain dynamics on the inter-organizational level, in *Proceedings of Third International Conference on Dynamic Modelling of Information Systems*, 1992, pp. 1-35.
- Teichroew, D., and Hershey, E., PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems, *IEEE Trans. Software Eng.* SE-3, 41-48 (1977).
- Walpole, R., and Myers, R., *Probability and Statistics for Engineers and Scientists*, Macmillan, New York, 1985.
- Ward, P., The Transformation Schema: An Extension of the Data Flow Diagram to Represent Control and Timing, *IEEE Trans. Software Eng.* SE-12, 198-210 (1986).

- Warren, J. R., CASE/Simulation Systems, Ph.D. Thesis, University of Maryland, Department of Information Systems, Baltimore, Maryland, 1992.
- Warren, J. R., and Stott, J. W., CASE/simulation: Making performance evaluation a normal part of IS development, in *Dynamic Modelling of Information Systems, II* (H. G. Sol and R. L. Crosslin, eds.), North-Holland/Elsevier Science Publishers, Amsterdam, 1992, pp. 219-250.
- Warren, J. R., Norcio, A. F., Stott, J. W., Canfield, G. C., and Freedman, R. W., The decision-support effectiveness of a simulation environment for information systems analysts: An exploratory study, in *Proceedings of Third International Conference on Dynamic Modelling of Information Systems*, 1992a, pp. 419-438.
- Warren, J. R., Stott, J. W., and Norcio, A. F., A prototype for including simulation of IS dynamics in CASE environments, in *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, 1992c, pp. 353-361.
- Warren, J. R., Stott, J. W., and Norcio, A. F., Stochastic Simulation of Information Systems Designs from Data Flow Diagrams, *J. Syst. Software* 18, 191-199 (1992c).

APPENDIX A. SAMPLE CASE

For each session, subjects are presented with a textual description of an organizational system with detailed descriptions of two of its subsystems. They also are presented with DFDs of each of the two subsystems and an associated data dictionary for each DFD. After perusing the materials, subjects are presented with eight dynamics questions regarding one of the two subsystems. These questions constitute a *case*. After a short break, subjects are presented with eight questions regarding the other subsystem. This constitutes the second case of the session.

Given below is the portion of the system description for a videotape rental system along with the detailed description of its return subsystem. Also given is the DFD for the return subsystem, a sample from the data dictionary of that DFD, and the eight dynamics questions for that case. The videotape rental cases (used for the first session) are based on an example given by Saldarini (1989). All four cases are given in their entirety in Warren (1992).

A.1 Systems Description

The videotape rental store currently uses a primarily manual information system in its dealings with customers. For each member, a membership envelope is maintained in a customer file. Whenever a member rents or returns videotapes, the appropriate membership envelope is pulled from the customer file and updated with information regarding the

transaction. When a customer rents tapes, the rental clerk fills out a rental agreement. Copies of the form are given to the customer, placed in the order book, and placed in the customer's membership envelope. When the customer returns the rented inventory, the customer presents the cope of the rental agreement and pays for the rental. The return clerk removes the copy of the rental agreement from the membership envelope and disposes of it, initials the cope in the order book, and initials and returns the customer's copy of the agreement to act as a receipt.

You should have a DFD and data dictionary entries for the return component of the videotape rental system. All entities in the return component are number 2x. Customers arrive at the return desk randomly (interarrival times following the exponential distribution) at an average rate of one every 4 minutes during the operating hours, noon to 8 p.m. weekdays. Customers present the return clerk with their pink customer copy of the rental agreement and payment along with the rental inventory. The rental clerk accepts these items and pulls each customer's membership envelope from the customer file (requiring a total of one minute, distributed exponentially). The clerk then performs the return processing, removing the gold copy of the rental agreement from the membership envelope and disposing of it, placing the payment in the cash drawer, returning the inventory to the shelving bin, pulling the appropriate white copy of the rental agreement from the order book, initialing it, returning it to the order book, and initialing and returning the customer's pink copy. This return processing requires a mean of 1.75 minutes, distributed normally with a standard deviation of 0.5 minutes. At the end of the day, the return clerk reshelves the inventory in the shelving bin, returning it to the stock shelves (requiring 0.3 to 0.7 minutes per customer return). After reshelving inventory, the clerk refiles the membership envelopes in the customer file (requiring 30 seconds per job, distributed exponentially).

A.2 Data Dictionary

Data dictionary entries annotated with dynamics information, such as this entry for process P2.1, are given for each DFD process, external entity, data store, and data flow corresponding to the elements shown in the DFD in Figure A1.

Process Entry

NAME:	P2.1: Look-up Membership
INPUT:	E2.1: Customer
	D2.1: Customer File

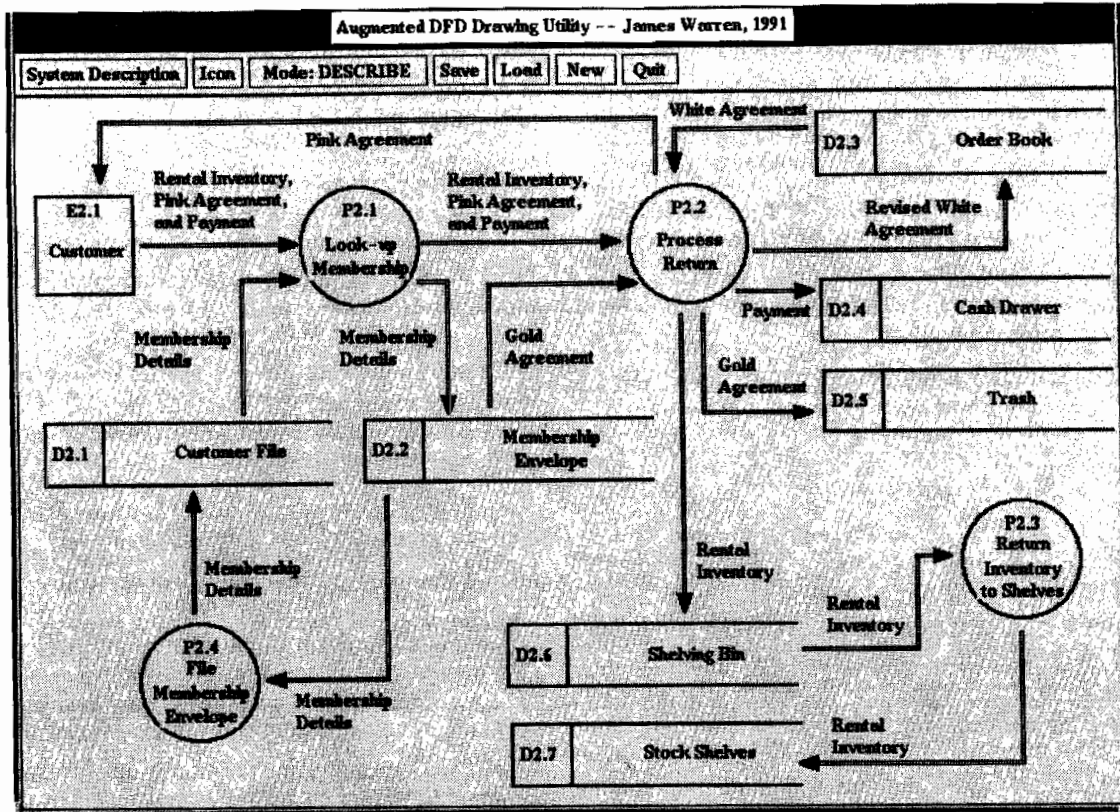


Figure A1. DFD of the return component of a videotape rental system.

OUTPUT: P2.2: Process Return
D2.2: Membership Envelope

JOB DURATION: 1.00 minutes/job, distributed exponentially. Processor waits for completion of job at output process before continuing to next job.

PROCESSORS: 1 unit: Mon-Fri 12:00-20:00 and idle

DESCRIPTION: Return clerk accepts inventory, form, and payment from customer and pulls their membership envelope from the customer file.

- at the return desk (excluding time waiting in line)?
- 5. How long does a customer spend at the return desk (including time waiting in line)?
- 6. How long is the line at the return desk (excluding the customer being served)?
- 7. How much time elapses between the time a customer's membership envelope is pulled from the customer file (at the beginning of customer service at P2.1) and the refiling of the envelope in the customer file?
- 8. How much time elapses between the time a customer begins service at the return desk (P2.1) and the placement of the returned inventory on the stock shelves?

A.3 Dynamic Questions

1. How many membership envelopes does the clerk refile at the end of each day?
2. How long does the clerk work each week at placing returned inventory on the stock shelves?
3. What is the utilization (in percent) of the return clerk at P2.1?
4. How long does a customer spend being serviced

Question 1 concerns throughput. This is expected-simple, because the question can be answered through simple multiplication (arrival rate × time period) and requires only the consideration of one queuing system. Question 3, concerning utilization, is expected-composite, because, although it can be solved using a simple set of multiplications and additions, it requires consideration of multiple

server/queues (P2.1 and P2.2). Questions 5 and 7 involve the system times of jobs. These are both emergent, because the system time is effectively intractable to calculate analytically, requiring use of simulation results. Question 5 is emergent-simple, because only process P2.1 (a single server/queue system) must be considered once simulation results are used. Question 7 concerns the processing sequence of customer files; information from multiple server/queue systems must be considered, and therefore it is emergent-composite.

APPENDIX B. ACCURACY MEASURES

The response to a dynamics question is used to obtain an accuracy measure between zero and one. The accuracy measure is

$$\text{score} = \left(1 - \left(\frac{\varepsilon}{\varepsilon + \min(\mathcal{R}, \hat{\mu})} \right) \right)^2 \quad (1)$$

where ε is a measure of the error in the subject's response, \mathcal{R} is the subject's response, and $\hat{\mu}$ is an estimate of the true value. For emergent-type questions, any response within the 95% confidence interval offered by the simulation environment for 10 replications of a 10-week simulation run^{A1} is considered perfectly accurate (assigned an accuracy measure of unity). For these questions,

$$\varepsilon = \begin{cases} 0 & \mathcal{R} - \hat{\mu} < \delta_{95} \\ \mathcal{R} - \hat{\mu} & \text{else} \end{cases} \quad (2)$$

where δ_{95} is the half-width of the 95% confidence interval of the estimated actual value and $\hat{\mu}$ is the best estimate of the actual value (the center of the confidence interval). For expected-type questions, ε is simply $\mathcal{R} - \hat{\mu}$ is the exact analytically determined correct response.

The observant reader may note that we are using the simulator itself to estimate the “correct” value, $\hat{\mu}$, for emergent-type questions. The simulator has been validated by comparison with other methods, such as analytical queuing models and comparison with popular simulation languages such as SIMSCRIPT. Given a valid stochastic simulator, the results will converge to the “true” values of the performance statistics for the given queuing network. Since a stochastic simulation always results in a range (rather than a point) estimate, we do not know the exact value for the queuing network statistics. The estimates used, however, are sufficiently

accurate to present no threat to the accuracy of the experiment. For instance, for Question 5 of the case presented in Appendix A, the estimated $\hat{\mu}$ is 6.18 with a δ_{95} of 0.2. If we apply the accuracy measure from equation 1 so as to compute an “accuracy score” for the outside of our confidence interval (putting in δ_{95} as the error, ε), then our 3% estimation error yields a score of ~ 0.94 , or a loss of 0.06 from a perfectly accurate response. Given that the observed treatment effect on emergent-type accuracy scores (see Section 4) is 1.13, we believe that the maximum likely errors are not sufficiently substantial to influence the outcome.

Three alternative formulations of accuracy scores are

$$\text{score}_1 = \left(1 - \left(\frac{\varepsilon}{\varepsilon + \min(\mathcal{R}, \hat{\mu})} \right) \right) \quad (3)$$

$$\text{score}_2 = e^{-\varepsilon / \min(\mathcal{R}, \hat{\mu})} \quad (4)$$

$$\text{score}_3 = e^{-2\varepsilon / \min(\mathcal{R}, \hat{\mu})} \quad (5)$$

Score₁ and score₂ are more “lenient” than the definition from equation 1, hereafter score₀; that is, these formulas return higher values for errors of equal magnitude (except for zero errors or zero responses, which result in accuracy scores of 1 and 0, respectively, for all four scoring methods). Score₃ is less lenient than score₀. The treatment effect is tolerant to alternative accuracy definitions, and the statistical significance of the treatment effect on emergent-type accuracy scores remains $\sim 0.04\%$ for all four definitions.

APPENDIX C. STATISTICAL ISSUES

C.1 Models and Hypotheses

We assess the treatment effect on accuracy scores by explicitly considering two effects in the experimental design:

- γ_1 : The improvement in accuracy from the first case of a problem-solving session to the second, that is, the intrasession learning effect.
- γ_2 : The improvement in accuracy from treatment level A to level B, that is, the treatment effect.

We developed a single score for each participant by adding the scores in those cases where the subject experienced treatment level B and subtracting from that sum the scores achieved in the cases performed at level A. That is, for a subject in ordering group 0, their score α_0 is:

$$\alpha_0 = s(t_2) + s(t_4) - s(t_1) - s(t_3) \quad (6)$$

^{A1} An arbitrary standard for an extensive simulation experiment.

where $s(t_i)$ is the sum of the accuracy scores achieved at time i . The theoretical range for α_0 is from -16 to 16 if scores from all eight questions of each case are used, or from -8 to 8 if only the four emergent-type questions of each case are considered (i.e., the emergent-simple and emergent-composite questions). α_0 can be thought to measure the combined effect of learning and treatment, $\gamma_1 + \gamma_2$.

For a subject in ordering group 1, their score α_1 is

$$\alpha_1 = s(t_1) + s(t_3) - s(t_2) - s(t_4) \tag{7}$$

α_1 can be thought to measure the treatment effect minus the learning effect, $\gamma_2 - \gamma_1$.

This leads to the following statistical null hypothesis:

$$H_0: \gamma_2 = (\alpha_0 + \alpha_1)/2 = 0 \tag{8}$$

This hypothesis is tested for the case where the $s(t_i)$'s are based on all eight dynamics questions, ranging from 0 to 8, and the case where only the emergent-type dynamics question are considered in the $s(t_i)$'s, thereby giving a range from 0 to 4.

Theoretically, the scores α_0 and α_1 are not normally distributed, because they have a limited possible range of values. In the pilot study, however, no observation approaches the limits of the range, and the statistical null hypothesis of normality is not rejected.

The model in equation 8 is exactly equivalent to testing the interaction between ordering group and time, where time has two levels: beginning of session and end of session. In other words, for each subject there are two scores, S_1 and S_2 , where

$$S_1 = s(t_1) + s(t_3) \tag{9}$$

$$S_2 = s(t_2) + s(t_4) \tag{10}$$

The interpretation of this model requires the assumption that the beginning-of-session versus end-of-session dimension is meaningful. A multivariate model that uses all four levels of time would be based on fewer assumptions about the time dimension, in the sense that the two-levels-of-time model discards some information about the correlations among the $s(t_i)$'s. The four-levels-of-time model, however, requires more statistical assumptions than the two-levels-of-time model, which reduces to a simple 2-by-2 design. In particular, because of the homogeneity of variance assumption, a four-levels-of-time model includes the assumption that the variance of all the $s(t_i)$'s is equal. Obviously, this entails

$$\sigma^2(s(t_1)) = \sigma^2(s(t_2)) \tag{11}$$

$$\sigma^2(s(t_3)) = \sigma^2(s(t_4)) \tag{12}$$

which are reasonable assumptions in light of the fact that the cases administered are randomized. There is, however, no randomization of cases between session 1 and session 2. Thus, the assumptions

$$\sigma^2(s(t_1)) = \sigma^2(s(t_3))$$

$$\sigma^2(s(t_1)) = \sigma^2(s(t_4))$$

$$\sigma^2(s(t_2)) = \sigma^2(s(t_3))$$

$$\sigma^2(s(t_2)) = \sigma^2(s(t_4))$$

are not well supported by the experimental procedure. The interpretation of a beginning-of-session versus end-of-session time dimension, which is well supported by the experimental design, constitutes less of a problem than incorporating the set of homogeneity assumptions into the statistical model.

In the pilot study (Warren et al. 1992a), subjects that performed a total of more than two simulation runs during the procedure have higher scores on emergent-type questions (summed over the two cases where simulation was available) than those who perform two or fewer simulation runs. This multiple simulation run effect is significant in the pilot study at the 5% level. Regression of accuracy scores versus number of simulation runs conducted is pre-planned in the current study. Regression is also used to analyze the relationship between accuracy scores and other usage variables. Except for testing of the multiple simulation run effect, the usage variable analysis is exploratory in nature. The intention is to build models for future experimental verification, rather than to test the statistical significance with which particular null hypotheses can be rejected.

C.2 Results

The treatment effect, γ_2 , is tested using both the two-levels-of-time model and the four-levels-of-time model (see Section C.1). The effect is tested for both overall and emergent-type accuracy scores.

Testing the two-levels-of-time model for overall accuracy scores results in the null hypothesis of no treatment effect being rejected with an F of 19.72 with 1 numerator and 19 denominator degrees of freedom, which is significant at the 0.03% level. The estimate for γ_2 is 2.195. For the two-levels-of-time approach, we must address the statistical assumptions of independence of observations, multivariate normality, and homogeneity of variance. Independence of observations is satisfied by the random assignment of participants to ordering groups. Testing the normality of the marginal distributions over time (beginning-of-session versus end-of-session) and ordering group with the Shapiro-Wilks test (SAS,

1989) reveals no significant violations of normality. Homogeneity of variance in each of the four cells (two levels of time by two ordering groups) is tested using Bartlett's test (Walpole and Myers, 1985), and the homogeneity of variance hypothesis is not rejected.

For the four-levels-of-time approach, treatment significance can be assessed with either a univariate or multivariate repeated measures model. Each model entails a statistical assumption beyond the model used for the two-levels-of-time analysis. For the multivariate approach, homogeneity of covariance matrices is assumed. For the univariate approach, there is the assumption of homogeneity of variance of treatment differences, for which sphericity is a sufficient condition (Stevens, 1986). For simplicity, the univariate approach is used.

Testing the normality of the marginal distributions reveals no significant violations of normality. Although Bartlett's test does reject the assumption of homogeneity of variance at the 5% level, there is an observed heterogeneity of variance (see Table 3). Maxwell and Delaney (1990) note that analysis of variance is robust to moderate violations of homogeneity of variance if group sizes are equal and not < 5 . When the groups with smaller variances have larger samples, the actual type I error rate is higher than the nominal rate. For these data, the group sizes are nearly equal (10 versus 11), and the degree of heterogeneity is moderate. From examples given by Maxwell and Delaney, the type I error rate should be inflated by much less than a factor of 2. Thus, univariate analysis proceeds with a cautious eye toward the type I error rate, but without transforming the data into a less interpretable form. Sphericity is violated by Mauchly's criterion (1.19% significance), therefore ϵ adjustment of the F static of the univariate repeated measures analysis of variance is appropriate (Maxwell and Delaney, 1990; SAS, 1989). The null hypothesis of no four-level time by order interaction is rejected with an F of 7.28 with 3 numerator and 57 denominator degrees of freedom, adjusted by a Huynh-Feldt ϵ of 0.8318, which is significant at the 0.08% level.

Using the more conservative four-level-of-time model, multiplying the test significance by two for each of 1) moderate violation of homogeneity of variance, 2) the use of two models (two and four

levels of time), and 3) multiple comparison (overall and emergent-type scores), yields rejection of null hypotheses of no treatment effect on accuracy scores at the 0.64% level.

In assessing the treatment effect on emergent accuracy scores for the two-levels-of-time approach, statistical assumptions are not violated. The null hypothesis of no treatment effect on emergent accuracy scores, formulated as a test of the interaction of two-level time with ordering group, is rejected with an F of 36.17 with 1 numerator and 57 denominator degrees of freedom, which is significant at the 0.01% level. The estimate for γ_2 is 2.2.

For the four-levels-of-time approach, treatment significance is assessed with a univariate repeated measures model. Testing of the multivariate normality assumption results in the null hypothesis of normality being rejected in most of the marginal distributions. Distributional transforms suggested by Stevens (1986) lead to the use of the arcsine-square-root transform to bring the emergent accuracy scores to normality. That is, for each subject i at time j , their emergent score E_{ij} is transformed to E_{ij}^* , where

$$E_{ij}^* = \sin^{-1} \left(\frac{E_{ij}}{4} \right)^{1/2}$$

The transform brings the scores into compliance with the normality assumption. The homogeneity of variance among the eight cells (four levels of time by two ordering groups) is not rejected. Sphericity is not violated by Mauchly's criterion (57.49% significance), therefore the null hypothesis of no four-level time by order interaction is rejected with an F of 15.48 with 3 numerator and 57 denominator degrees of freedom, which is significant at the 0.01% level. Incidentally, essentially identical results are obtained using the untransformed scores, casting doubt on the necessity of using a transform.

Multiplying the test significance by two for each of 1) use of two models (two and four levels of time), and 2) multiple comparison (overall and emergent-type scores) yields rejection of the null hypothesis of no treatment effect on emergent-type scores at the 0.04% level.

Further details of the choice of models and the data analysis can be found in Warren (1992).