Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

REAL TIME VALIDATION OF MAN-IN-THE-LOOP SIMULATIONS

William C. Hopkinson

Department of Industrial Engineering
University of Central Florida
Orlando, Florida 32816, U.S.A

José A. Sepúlveda

Department of Industrial Engineering
University of Central Florida
Orlando, Florida 32816, U.S A.

ABSTRACT

The objective of this research is to develop a methodology which allows the user to validate the decisions of the man in the loop dynamically in real time by reducing type II errors. Case-based reasoning provides a modeling paradigm for designing a system which can aid in the validation process. This paradigm provides for a mechanism to capture cases that can be used in the validation process. A proof of principle demonstration is being developed that operationally validates the decisions made by a military commander in a high resolution simulation. This demonstration will establish the ability of an expert system to operate in real time and assist in the validation process.

1 INTRODUCTION

Simulations, as a tool to answer various questions in both commercial and military areas, have proliferated in the past decade. They are no longer just used by the "computer expert" or "operations research specialist" but rather by novice, non-simulation experts needing answers to critical questions within time constraints. Man in the loop simulations are a class of computer models which are typically used for training students in command type of decisions and are distinguished by the decision making process being controlled dynamically by the trainee. Quality control of computer simulations is controlled through verification and validation throughout the life cycle of the model. Verification is the process that determines that the model and simulation functions as it was originally conceived, specified, and designed. Validation is the process that addresses the credibility of the model and simulation in its depiction of the modeled world. Computer simulations are a tool used to evaluate a model numerically in order to learn some insight about a particular phenomena of interest (Law and Kelton 1991). The objective of a computer simulation may vary from providing analytical insight to a problem to serving as a training device for students. There are varying types of computer simulations ranging from discrete event, continuous event, to intelligent based simulations.

Computer simulations, as a tool to answer various questions in both commercial and military areas, have proliferated in the past decade. They are no longer just used by the "computer expert" or "operations research specialist" but rather by novice, non-simulation experts needing answers to critical questions within time constraints. Tools are being developed which will allow the user to develop a simulation model with little or no in-depth knowledge of the process required to develop a quality model. This proliferation of simulations opens up the problem of misusing a simulation with the possibility of making critical decisions based on incorrect information.

Perhaps a more glaring problem in the simulation arena is the total confidence users often place upon the results obtained from a simulation run. Quality control of simulations in terms of verification, validation, and accreditation (VV&A) is often overlooked if not totally ignored in the development and employment of simulations. However, it is recognized as a critical aspect of the modeling process. In a recent survey of modeling experts (Willemain 1994), the validity of a simulation model was ranked as the most important quality of an effective model.

In the training arena, a particular class of simulation model is being developed which offers some unique challenges to the modeler in terms of validation. Man in the loop simulations are characterized by an interactive mode of operation between the player (or trainee) and the simulation. This interaction provides an added dimension to the uncertainty in the operation of the model. The decision making element provided by the human creates an environment for anomalies and errors resulting from player/simulation interaction. Errors can range from incorrect inputs (i.e., wrong key stroke for a data entry) to invalid decisions by the player (i.e., a player using an object in the simulation outside of its normal intended purpose). Examples of man in the loop simulations include power plant gaming tool for training power plant operators; disaster planning simulations for response of government agencies to catastrophic events; and military wargaming simulations used to train future battlefield commanders. The focus for this discussion will be on the decision making aspects of man-in-the-loop simulations and how to validate those decisions: How does the modeler ensure or validate the decisions being made by the player in a man-in-the-loop simulation without placing unrealistic restrictions on the gamer.

2 VERIFICATION, VALIDATION, AND ACCREDITATION DEFINED

The terms verification, validation, accreditation, and evaluation have often confusing and overlapping meanings depending on the user. The software engineer views these concepts in terms of the quality of the software lifecycle development process. The simulation modeler views VV&A in terms of the phenomena being modeled. The modeler is most concerned with how accurately and with what degree of precision does the developed computer simulation model represent the real world.

Verification is the process that determines that the model and simulation functions as it was originally conceived, specified, and designed (Department of the Army 1993). The IEEE Standard Glossary of Software Engineering Terminology defines verification as "the process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase" (IEEE 1984). The objective of verification is to obtain a correct computer implementation of the developed conceptual model. Typical techniques used during the verification process include top-down design, structured programming, and program modularity (Balci and Sargent 1984). Verification can be thought of in terms of two basic parts; logical and mathematical verification and program (or code) verification. Logical and mathematical verification checks basic algorithms and rules to ensure the implementation is as intended by the designer. Examples of these types of errors include division by zero, incompletely specified logic, or nonsensical results when certain variables take on extreme values. Program verification ensures that the model representations are correctly implemented in the computer program. An example of program verification includes discovering and correcting typographical errors (Davis 1992). In summary, verification is primarily concerned with ensuring the developed model has been correctly implemented.

Validation is the process that addresses the credibility of the model and simulation in its depiction of the modeled world (Department of the Army 1993). Validation addresses the question does the model accurately represent the real world phenomena within an acceptable level of confidence for the intended use of the simulation. This breaks the process of validation into two parts; accurate representation and acceptable level of confidence. These two parts represent a difficult nature of the validation process in establishing the acceptable criteria for accuracy and confidence (Davis 1992). Determining these criteria is a management issue which the modeler and the customer must agree on prior to development and continually monitor during the lifecycle of the simulation.

Validation focuses on the credibility of the model and simulation in its depiction of the modeled world. In a man-in-the-loop simulation, the man becomes an embedded part of the model (the model encompassing the entire phenomena being examined). The decision making aspect of the human, as a part of the model, must be examined in order to validate the model as a complete entity. It is these human decisions that are the focus of this methodology. The concept of the man's decision making aspects being part of the modeled system may be open to debate. However, if the man makes no decisions throughout the entire process (understanding that the decision to do nothing is still a decision) in a man-in-the-loop simulation, then the model will not function as designed.

Similarly, data validity must be addressed in a closed simulation to ensure the computer simulation model is receiving valid inputs into the developed algorithms. Knepell and Arangno (1993) specifically address the difficulties associated with validation of man-in-the-loop simulations and the need to validate the role the human plays in the simulation. They discuss operational validation, pertaining to man-in-the-loop simulations, as examining "the model performance over the range of model inputs for the intended use". They further state "operational validation testing becomes more complex than for computer-based models due to the addition of the human" and that "man-in-the-loop model operational validation testing emphasizes the areas of model support of the human element of the model". They consider the human interaction to be an integrated part of the model and must be addressed in terms of model validation. For purposes of this research, the human decision making is considered part of the model and therefore validation of the decisions must be explored within the context of a full validation process. Validation is the most difficult aspect of the

	H_0: Model valid	H_1: Model not valid
Accept model	OK	Type II Error (β)
Reject model	Type I Error (α)	OK

Figure 1: Validation Outcomes

VV&A problem. Modelers must balance the need for accurate representation of the phenomena against the available resources to solve the problem. This research will speak to the need for addressing this balance within man in the loop simulations.

Accreditation is the final aspect of the simulation quality control problem. Accreditation is a term primarily used in the Department of Defense arena and refers to using the correct simulation to answer the question being asked (Department of the Army 1993). Typically, accreditation is a managerial issue in which a study director reviews the issues in a particular study and "accredits" or approves a simulation as being the correct tool to answer the questions.

2.1 Errors Associated with Simulation Validation

The types of possible outcomes for a validation exercise can be discussed in terms of hypothesis testing. There are two types of errors associated with hypothesis testing (Sargent 1985). These types of errors are the model builder's risk (α) or type I error and the model user's risk (β) or type II error. Figure 1 depicts the possible outcomes of a model validation effort. If a model is rejected as being valid when in fact it is (type I error), the modeler is the party who is damaged in that an acceptable model is not used. If a model is accepted as valid when in fact it is not (type II error) the customer or user is placed at risk. For a military commander learning about the effects of a tactical decision during an operation, acceptance of an invalid model may result in teaching the user the incorrect lesson. In model validation, model user's (i.e., the customer) risk is extremely important and must be kept to a minimal in order to ensure customer satisfaction and confidence in the simulation model. The focus of this paper is to explore a new methodology for reducing type II errors.

3 VALIDATING DECISIONS

How to validate the decisions of the human within a man in the loop simulation is a multi-dimensional process and may be defined within the following framework:

1. determining the intended goal/mission for the human
2. interpreting the input of the human
3. comparing the input of the human with the established goal/mission
4. responding to the comparison in terms of prompting the user or adapting the simulation.

These four steps provide the structure for conducting an operational validation evaluation of the decision making process for a man in the loop simulation. The intended mission/goal for the human is an a-priori goal provided to the trainee for each exercise. The goal typically employs a series of sub goals which the trainee must negotiate in order to achieve the provided goal.

Step two, interpreting the input of the human into the simulation, requires an understanding of the human general problem solving process (Newell and Simon 1972). The assumption is made at this time that the trainee will attempt to solve the problem (reach the intended goal) using a logical problem solving method. Illogical methods to solve the problem will not be viewed as a valid approach to the problem. The difficult aspect of the problem is how to reduce the number of possible combinations for attaining the desired goal. This problem has a combinatorial explosion aspect which must be addressed through the use of planning methods.

Comparison of the humans inputs with the intended goal (and sub goals) represents the actual validation of the decision. The input from the human must be interpreted and then compared against the acceptable (or unacceptable) solutions using both qualitative and quantitative techniques. This comparison will determine the operational validity of the decision made by the human.

The final aspect of the problem is having the simulation model respond to the comparison made in step three. The response may be either active, semiactive, or passive. An active response would provide some method for prompting the trainee that the input is valid or invalid following each input. A semiactive response would prompt during an invalid decision and allow the trainee to alter the decision. A passive system would record the decisions and allow the user to continue the game to its conclusion and then use the invalid decisions as an instruction point.

The constant throughout all previous validation/ evaluation efforts is that a static approach has been taken. Statistical approaches, such as confidence intervals and hypothesis testing have relied on conducting a test at the conclusion of the simulation and making a decision as to the obtained results were acceptable or validated through some domain. In the training environment, a controller often acts as the "umpire" for a particular scenario to determine or evaluate the validity of the decisions made by the human in the loop (Knepell and Arango 1993). This is a highly subjective evaluation and is manpower intensive requiring an expert to be present during the exercise.

Typically, if a user is able to achieve the goal/mission state the exercise is considered successful. However, it is possible to arrive at the final objective while not remaining within the "sandbox" of the scenario. The journey to the goal must be evaluated against intermediate objectives to ensure the users decisions were valid. A methodology capable of interpreting the humans input to the simulation, determining appropriate intermediate goals, and comparing/evaluating the goals with the human input would provide an increased level of "believability" to the results obtained during a training session.

3.1 Proposed Validation Methodology

The methodology developed focuses on reducing the models users risk by developing a system that identifies invalid decisions made by the trainee. In a complex man-in-the-loop simulation where free play is encouraged (such as a military simulation) the number of valid decisions a trainee may make is infinite. To attempt to develop a system that incorporates all the possible valid decisions would be impractical. However, the number of invalid decisions may be significantly less and thus greatly reduce the search space for comparison of decisions.

Focusing on the reduction of type II errors (i.e., model user's risk), the methodology proposed validates the decision by the gamer at varying times during the simulation (rather than at the end of a run). In actuality, the output of the validation process is not saying the input by the gamer was valid but rather saying the input does not match or coincide with a known invalid solution. The converse of this method would be to attempt to validate all inputs as being acceptable. This approach would not be feasible in a computer simulation that provides for free play and promotes creativity. If the simulation model being explored had a limited domain with a relatively small solution space, then the concept of validating all feasible inputs may be possible. However, as the simulation model becomes more complex and the solution space increasingly large, the possible combinations for valid inputs becomes increasingly difficult to handle. The question to the approach of exploring invalid solutions in complex domains then becomes is there value in knowing that a particular response or input to a simulation model does (or does not) match a previous case. Figure 2 is a depiction of the validation cycle for the decision making aspects for a man-in-the-loop simulation.

The operational validation cycle describes the continuous cycle of validation throughout the simulation. The cycle can be discussed in terms of gamer input, situational assessment, comparison and reply. The gamer provides an input in terms of an operational decision in the man-in-the-loop simulation. This input is then transformed into a situational assessment which allows for comparison with the case library. If a match with an invalid solution previously captured in a case is determined, a response to the gamer is provided. If there is no match, the simulation continues. Any case that does not may be stored for an after action review. This review allows for determination following the simulation run if any decisions made during the game should be considered invalid. If following the after action review it s determined that a particular strategy employed is invalid, the stored case can be added to the case library. This creates a system capable of constantly improving.

3.2 Case-Based Reasoning

Case-based reasoning is an approach to problem solving which adapts previous solutions to a given problem to solve a new problem. This method of problem solving differs from rule-based reasoning which chains rules of inference together to solve a problem (Riesbeck and Schank 1989). Typically a case-based reasoner finds those cases in memory that solved problems similar to the current problem and adapts the previous solution or solutions to fit the current problem, taking into account any difference between the current and previous situations. A case-based reasoner has a case library. Each case describes a problem and a solution to that problem. The reasoner solves new problems by adapting relevant cases from the library.

Case-based reasoning is an alternative to rule-based reasoning. A rule-based reasoner has a large library of rules of the form If A then B. These rules are chained together in various combinations to solve problems. Rule-based systems are flexible and produce nearly optimal answers but are typically slow and can be prone to errors. A case-based reasoner will be restricted to variations on known situations and produce

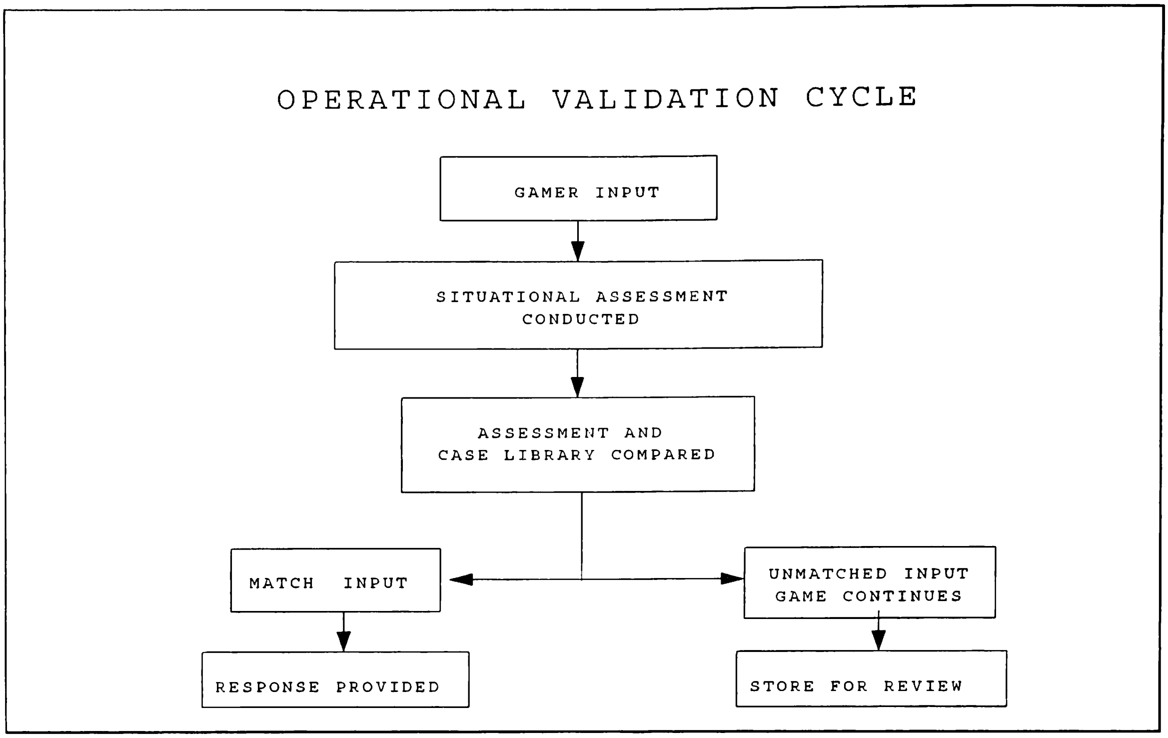

Figure 2: Operational Validation Cycle

approximate answers, but it will be quick and its answers will be grounded in actual experience. In a limited domain problem such as an advisor, a rule-based system would be the preferred system. However, as the complexity of the domain increases, case-based reasoning provides a considerable advantage to a rule-based approach. Rule-based reasoning results in long chains of rules for complex problems. Case-based reasoning provides a short connection between the input case and the retrieved solution. There are two main advantages of case-based reasoning over rule-based reasoning. First, expertise is more like a library of past experience than a set of rules. Second, complex problems typically are too involved to be solved using a rule-based structure within a reasonable amount of time. The primary weakness for case-based reasoning is the development of cases to support the system. The sparser the case-base, the more reliance that must be placed into the adaptive strategy.

3.3 Implementation of CBR in the Validation process

Case-based reasoning is being implemented into a military man-in-the-loop simulation to demonstrate the feasibility of validating the decisions made by the commander or trainee during a training exercise. At Figure 3 is the architecture for the proposed system. The Operational Validation Evaluation Response (OVER) analyzer will serve as the vehicle for demonstrating the ability to operationally validate the input decisions of the trainee in real time. The OVER analyzer module is a case-based reasoning expert system which will run passively in the background of the simulation model JANUS (Department of the Army 1992). Inputs from the trainee will be processed through an interface to the working memory of the system. The Janus Enhanced Data Analyzer (JEDA) is a high-speed data analyzer which allows for real time access to the output data files during execution of the simulation.

In developing the OVER analyzer, the strategy to be used is the reduction of type II errors i.e. the model users risk. A type II error is the acceptance of an input (and in fact the simulation model) to be valid when it is in fact not. In order to reduce the type II error, the OVER analyzer will look at identifying incorrect tactics or strategies and then alerting the trainee of the incorrect strategy. A series of sentinel rules that can not be violated will initially be developed. An example of a sentinel rule for a military simulation may be a particular system may not be employed using a particular tactic (i.e. a nuclear device may not be employed against a single platoon). These sentinel rules serve as the initial building block for the OVER analyzer. The number of possible valid strategies for a particular military tactic is infinite and too large to develop in a knowledge base. However, there are numerous invalid strategies which are known that could

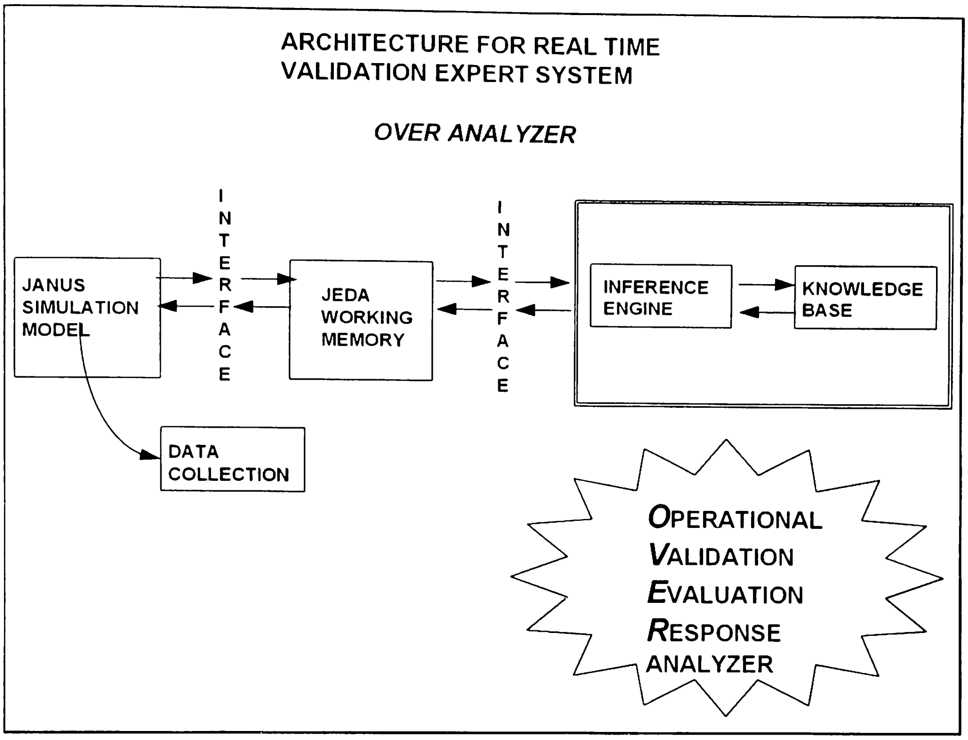

Figure 3: OVER Analyzer Architecture

be developed. By identifying these invalid strategies, the type II error for the simulation can be significantly reduced. Cases are then developed that represent known invalid strategies. As more games or runs of the simulation are conducted, the number of cases for invalid tactics or strategies can be identified and included in the knowledge base.

This approach to validation greatly reduces the type II error. By conducting this validation in real time, the confidence a trainee can have in the results or training experience from the simulation is greatly increased. While invalid strategies may be used during a game, if the result of the strategy is a poor outcome, a new case can be developed from this invalid strategy for use with future simulation runs. This will result in a simulation that is constantly improving and validating its operational decisions during real time. If an invalid tactic or strategy is employed which allows for a favorable outcome, a review of the strategy must be conducted to review the technique.

4 SUMMARY

The objective of this paper is to discuss the development of a new methodology for operationally validating a man-in-the-loop simulation in real time. This approach is currently under development and will be demonstrated in a military man-in-the-loop simulation. Case-based reasoning will be employed to develop invalid cases to be compared to the inputs of the trainee to validate the decisions. This approach will reduce the type II error while developing a better knowledge base. This dynamic approach to validating a simulation will increase the confidence users have in the quality of the simulation model.

REFERENCES

Balci, O., and Sargent, R. 1984. Validation of simulation models via simultaneous confidence intervals. *American Journal of Mathematical and Management Sciences*:374-406.

Banks, J., Gerstein, D., and Searles, S. 1987. Modeling process, validation, and verification of complex simulations: a survey. In *Methodology and Validation Proceedings of the Conference on Methodology and Validation, 1987,* ed. O. Balci, 13-18. The Society of Computer Simulation, San Diego, California.

Davis, P.K. 1992. Generalizing concepts and methods of verification, validation, and accreditation (VV&A) for military simulations. RAND/R-4249-ACQ, The RAND Corporation, Santa Monica, California.

Department of the Army 1993. Army Regulation 5-11, *Models and Simulations.*

Department of the Army 1992. User's Manual for JANUS 3.X Unix Version. prepared by Titan Corp for TRADCOC Analysis Command, Fort Leavenworth, Kansas.

Hayes-Roth, B., and Hayes-Roth, F. 1978. Cognitive process in planning. RAND/R-2366-ONR, The RAND Corporation, Santa Monica, California.

IEEE Standard Document No.830-1984.

Knepell, P.L., and Arangno, D.C. 1993. *Simulation Validation A Confidence Assessment Methodology.* Los Alamitos, California: IEEE Computer Society Press.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, California: Morgan Kaufmann Publishers.

Law, A.M., and Kelton, W.D. 1991. *Simulation Modeling and Analysis.* New York: McGraw-Hill.

Riesbeck, C.K., and Schank, R.C. 1989. *Inside Case-Based Reasoning.* Hillsdale, New Jersey: Lawrence Erlbaum Associates Publishers.

Sargent, R.G. 1985. An expository on verification and validation of simulation models. In *Proceedings of the 1985 Winter Simulation Conference,* ed. D. T. Gantz, G. C. Blais, and S. L. Solomon, 15-21. Institute of Electronics Engineers, San Francisco, California.

Willemain, T.R., 1994. Insights on modeling from a dozen experts. *Operations Research* 42:213-222.

AUTHOR BIOGRAPHIES

WILLIAM C. HOPKINSON is a Major in the United States Army pursuing a Ph.D. in Industrial Engineering at the University of Central Florida. He received his Masters in Operations Research from the Naval Postgraduate School, Monterey, California. His research interests are in the areas of military simulation, and verification and validation of computer simulations.

JOSÉ A. SEPÚLVEDA is Director of Graduate Affairs in the College of Engineering at the University of Central Florida, Orlando, Florida. He received an Ingeniero Civil Químico degree from the Universidad Santa María, Valparaíso, Chile, and MSIE, MPH, and Ph.D. (Industrial Engineering) degrees from the University of Pittsburgh. Dr. Sepúlveda is a registered Professional Engineer in Florida and Chile. He is a member of the Institute of Industrial Engineers, the Society for Computer Simulation, INFORMS, the American Public Health Association, and the Hospital Information Management Systems Society. His major areas of research interest are measuring and modeling training effectiveness; object oriented programming and expert systems applications in simulation; tasks scheduling in complex and risky enviroments; faster than real time simulations; integration of an intelligent tutoring component into a team training device; intelligent simulations applications in training shift supervisors; and applications of hypermedia in Emergency Room environments. He has written two books and numerous publications.