Ling Rothrock · S. Narayanan *Editors*

Human-in-the-Loop Simulations

Methods and Practice

Springer

Human-in-the-Loop Simulations

Ling Rothrock · S. Narayanan
Editors

Human-in-the-Loop Simulations

Methods and Practice

Springer

Ling Rothrock Ph.D.
The Harold and Inge Marcus
Department of Industrial and
Manufacturing Engineering
The Pennsylvania State University
210 Leonhard Building
University Park
PA 16802
USA
e-mail: lxr28@engr.psu.edu

S. Narayanan Ph.D.
405 Russ Engineering Center
Wright State University
3640 Col. Glenn Hwy
Dayton
OH 45435
USA
e-mail: s.narayanan@wright.edu

ISBN 978-0-85729-882-9 e-ISBN 978-0-85729-883-6
DOI 10.1007/978-0-85729-883-6
Springer London Dordrecht Heidelberg New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Both the editors of this book were exposed to human-in-the-loop simulations while pursuing their doctoral degrees in the Center for Human-Machine Systems at Georgia Tech. In fact, S. Narayanan served as Ling Rothrock's teaching assistant for the simulation course taught by Prof. Christine Mitchell. It has been over 15 years since our Georgia Tech days and we have both been continuously active in human-in-the-loop simulation research. The purpose of this book is to leverage the lessons we learned to provide researchers and teachers with a handbook on how human-in-the-loop simulations can be used to study human interactions in various contexts.

A human-in-the-loop (HITL) simulation is a modeling framework that requires human interaction. Traditional simulation studies regard human interaction as an external input to the system being considered. However, studies of complex systems in today's technological landscape must include humans as active participants. For example, a study of highly automated call centers must include human judgement and decision making and the accompanying task context. The emergence of HITL technologies, therefore, enables researchers and practitioners to investigate the complexities of human-involved interactions from a holistic, systems perspective. The handbook consists of contributed chapters from experts in academia and industry in the area of human-in-the-loop simulation. By reading it, the reader should gain an understanding of what an HITL simulation is and how it differs from traditional simulations, an appreciation for how HITL simulations can be used to study human involvement in complex systems, and an understanding of the current research thrusts involving HITL simulations.

The first section of the book consists of three foundational chapters to introduce HITL simulations. In Chap. 1, S. Narayanan and Phani Kidambi provide an overview of the history, features, and trends of HITL simulations. In Chap. 2, Ling Rothrock discusses the human performance measurement and evaluation of HITL simulations using a temporal logic framework to represent windows of opportunity. In Chap. 3, Michael Matessa and Walter Warwick describe a graph-based interface language called GRBIL that facilitates interaction between human operators and the task environment within an HITL simulation.

The second section of the book consists of three chapters that use HITL simulations to study cognitive models. In Chap. 4, Hui Xi, Seungho Lee, and Young-Jun Son discuss the requirements for an HITL simulation to validate an integrated model of human behaviour at both the tactical and strategic levels of decision making. In Chap. 5, Frank Ritter, Michael Schoelles, Karen Quigley and Laura Klein assess the need for HITL simulations to inform cognitive model development. In Chap. 6, Anand Tharanathan, Paul Derby, and Hari Thiruvengada present a study of metacognition using HITL simulations.

The third section of the book contains two chapters that describe human-in-the-loop processes for complex simulation models. In Chap. 7, Timothy Simpson, Dan Carlsen, Matthew Malone, and Joshua Kollat present a human-in-the-loop process to guide simulations which explore trade spaces to optimize engineering designs. In Chap. 8, Dhananjai Rao, Alexander Chernyakhovsky, and Victoria Rao demonstrate the use of human-in-the-loop to guide bio-simulations to conduct epidemiological analyses.

The fourth section of the book consists of four chapters that characterize the use of HITL simulations addressing specific problems in particular contexts. In Chap. 9, Michael Hass, Robert Mills, and Michael Grimaila use HITL simulations to increase understanding of the potential effects of cyber-attacks in order to guide the development of contingency planning. In Chap. 10, Subhashini Ganapathy, Sasanka Prabhala, S. Narayanan, Raymond Hill, and Jennie Gallimore use HITL simulations to compare the effectiveness of a human-computer integrated routing application against an automated planner for the military. In Chap. 11, Hari Thiruvengda, Anand Tharanathan, and Paul Derby use HITL simulations to train cognitive skills required for military operations. In Chap. 12, Sasanka Prabhala, Jennie Gallimore, and Jesse Lucas use HITL simulations to assess the impact of different levels of automation on human control of semi-autonomous systems.

We expect the book to be of use to engineers interested in advancing the design and implementation of test beds to investigate human-machine interaction, to psychologists seeking to understand human judgment and decision making in dynamic task environments, and to computer scientists interested in building hybrid systems to facilitate human–machine cooperation.

Ling Rothrock
S. Narayanan

Contents

Chapter 1
Interactive Simulations: History, Features, and Trends

S. Narayanan and Phani Kidambi

Abstract Interactive simulations are well suited for analyzing large and semi-structured problems, especially in which human interaction is an important consideration. Interactive simulations, also known as human-in-the-loop simulations, can be used for generating a better understanding of human behavior under complex situations by visually highlighting features that may not be readily accounted for in traditional simulations. Additionally, they can also be used for systems analysis under operational conditions as well as for simulator-based training. In this chapter, we discuss the differences between interactive simulations versus traditional and animated discrete-event simulations; present a brief historical overview of interactive simulations, highlight the architectural features, and summarize the application and development trends.

1.1 Human-in-the-Loop Simulations Versus Other Types of Simulations

The fundamental objective in computer simulation is to develop simplified software abstractions to represent the behavior of a complex system over time. Interfaces to simulations convey the dynamic behavior of the model to the analyst.

This chapter is adapted from the journal article Narayanan S et al. (1997) with permission from SAGE publications.

S. Narayanan · P. Kidambi (✉)
College of Engineering and Computer Science, Wright State Reserch Institute,
405 Russ Engineering Center, Wright State University, Col. Glenn Highway 3640,
Dayton, OH 45435, USA
e-mail: phani.kidambi@wright.edu

S. Narayanan
e-mail: s.narayanan@wright.edu

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_1, © Springer-Verlag London Limited 2011

Traditionally, simulations have been classified into discrete-event versus continuous simulations, and within discrete-event simulations into event scheduling-based methods versus process interaction methods. It is useful to also examine simulations from the perspective of discrete-event simulations versus animated simulations and interactive or human-in-the-loop simulations.

Table 1.1 presents a comparison of interactive simulations with traditional discrete-event simulations and animated simulations along the following dimensions: nature of suitable problems, simulation development life cycle, time transition of simulation clock, nature of user interaction, role of the graphical interface, types of output analysis, and example of software packages for each category.

Interactive simulations are well suited for large and semi-structured problems, especially in which human interaction is an important consideration. These problems feature situations that cannot be easily predicted. An example of such a situation is human decision making behavior in scheduling and planning in job shop manufacturing systems (Hurrion and Secker 1978; Dunkler et al. 1988). Interactive simulations can be applied in the decision support or simulator perspective under those situations (O'keefe 1987). Human-in-the-loop simulations can result in a better understanding of human behavior under complex situations. Interactive simulations can visually highlight features that may not be readily accounted for in traditional simulations. For example, Bell and O'Keefe (1987) illustrate an interactive simulation of a complex rail locomotive system that highlighted realistic crisis situations to users. In that system, it was virtually impossible to build a programmed response to every possible crisis situation, and therefore an interactive simulation was an effective alternative.

Interactive simulation development is different from the traditional simulation life cycle as the specification of interaction and animation is concurrent with model specification. The level of detail in the interface can, however, vary depending on the purpose of the modeling exercise. Graphical interfaces in interactive simulations depict dynamic system states, highlight performance measures, and contain interface objects that accommodate command line inputs and other user interaction. Output analysis in interactive simulations primarily involves transient systems analysis because human interaction at different times during the simulation prevents the simulation from reaching a steady state.

1.2 Historical Perspective of Human-in-the-loop Simulation Development

Concurrent with software technologies development, the field of interactive simulations has evolved. Fig. 1.1 illustrates the historical development in the field of interactive simulations from the viewpoint of the interface structure, programming languages predominantly applied, and the associated programming paradigm in system implementation.

Table 1.1 Comparison of interactive simulations with traditional discrete-event and animated simulations

Topic/issue	Traditional discrete-event	Animated discrete-event	Interactive
Nature of suitable problems	Well-structured problems	Well-structured to semi-structured problems	Semi-structured to unstructured problems
	Small- to medium-scale systems	Medium-to large-scale systems	Small-to medium-scale systems
	Human interaction not a critical consideration	Human interaction not a critical consideration or can be captured completely	Human interaction is a critical consideration
Simulation development life cycle	Traditional	Traditional with animation specification following simulation model configuration	Interface and interaction specification concurrent with simulation model specification
Time transition of simulation clock	From one discrete-event to another	From one discrete-event to another with scaled time for animation	Discrete-event to another
			Scaled time
			Real-time
User interaction	None	Little interaction, may be able to alter simulation experimental parameters	Able to alter simulation experimental parameters
			Able to alter simulation system dynamics
Graphical interface	None	Displays dynamic system states	Displays dynamic system states
		Displays performance measures	Displays performance measures
			Interface objects accommodate command line input and other user commands
Output analysis	Steady state or terminating simulation analysis	Steady state or terminating simulation analysis	Transient systems analysis
			Human factors engineering analysis methods
Software	Simulation language (e.g., GPSS V)	Simulation packages with animation capabilities (e.g., SLAM/TESS, SIMAN/CINEMA)	Simulation packages (e.g., VISION, SEE-WHY, WITNESS)
	Programming language (e.g., FORTRAN)	Programming Language (e.g., FORTAN, C)	Programming language (e.g., C, C++, Smalltalk)

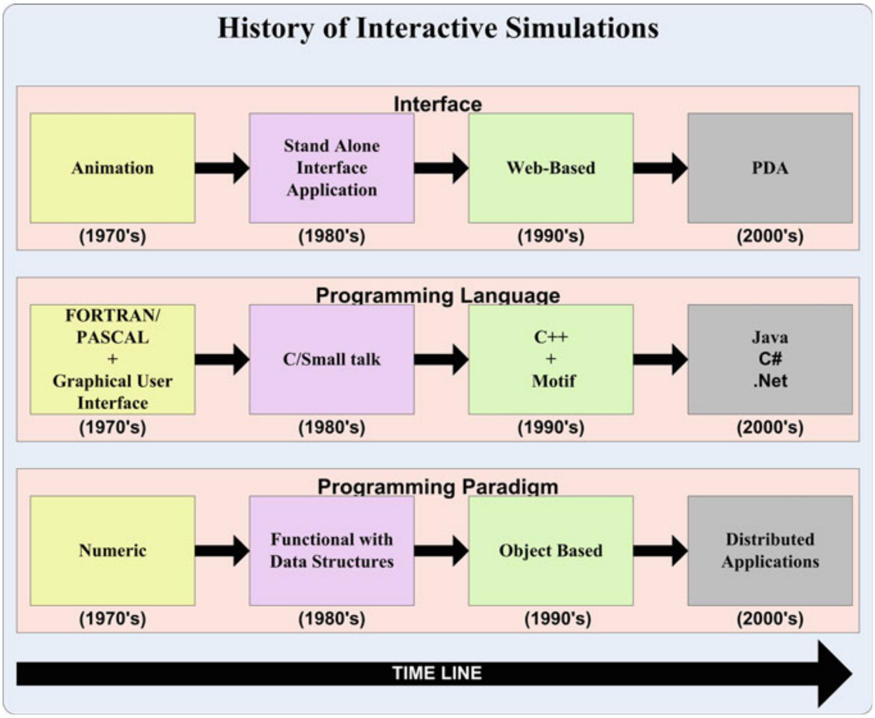


Fig. 1.1 Historical development of interactive simulations

1.2.1 Interface

1.2.1.1 Animation

Animating simulation output through the graphical interface was perhaps the first step towards the adoption of human-in-the-loop simulation. In animation, the fundamental goal is to portray the simulation output. The early focus was on displaying the simulation output and had little input from the simulation analyst during the execution. CINEMA (Pedgen et al. 1985), TESS (Musselman 1986), Xcell (Conway et al. 1985) are some of the early simulation animation tools.

1.2.1.2 Stand-Alone Interfaces

With the advances in computing power and graphical user interfaces, there was an increasing interest in the area of visual interactive simulations during the early 1980s and 1990s (Bell and O'Keefe 1987; Bell 1991; Hurrion 1980; Lyu and Gunasekaran 1993). In visual interactive simulations (VIS), interfaces serve not only to display the state of the simulated system, but also to allow an analyst to interact with the executing simulation. As the simulation executes in real-time or scaled time, the analyst can modify the parameters and alter the dynamics of the simulated system.

The VIS approach offers several potential advantages. First, it allows the user to interactively make complex decisions. For example, Hurrion and Secker (1978) found that the rules used by human schedulers in job shop scheduling were difficult to encapsulate in simulation models. VIS offered a viable alternative by allowing complex decisions to be made externally. Second, VIS is useful in studying the effectiveness of real-time, human decision making in complex systems. Dunkler et al. (1988), for example, used an interactive simulation of a flexible manufacturing system and compared the effectiveness of various automatic scheduling strategies with that of human scheduling in expediting parts through the system. Third, the display of the simulated system in VIS can be visually appealing and can increase effective communication between a manager and the simulation analyst in model development (Bell 1991; Bishop and Balci 1990). Fourth, the dynamic visual representation in VIS can highlight logical inconsistencies in the model and can therefore be effective in model verification and validation. Finally, since the user of VIS actively participates in the execution of the simulation, there is potential for increased user confidence in applying the results of the simulation (Kirkpatrick and Bell 1989).

There are two major challenges with the VIS approach (Bell and O'Keefe 1987; Bishop and Balci 1990; Paul 1989). First, due to human interactions during the execution of the simulation, the simulation experiments are hard to duplicate and the output is not readily amenable to traditional statistical analysis. Second, a user interacting with the simulation may observe a snapshot of the system and may prematurely conclude that the system will always exhibit the observed characteristics without the benefit of detailed statistical analysis.

Despite the problems outlined above, interactive simulations are often beneficial in the analysis of complex, dynamic systems. O'Keefe (1987) outlines the different perspectives of VIS: statistical, decision support, and simulator. Under the statistical view, there is little or no user interaction with the simulation model during program execution. The interfaces under this mode are primarily for post-simulation animation or performance analysis (Bishop and Balci 1990). Under the decision support perspective, emphasis is placed on "what–if" analysis by the user. A user can evaluate alternate scenarios through interaction with the simulation model. The interaction can be user-initiated or model-prompted. Prompting the user to make a scheduling decision is an example of model-prompted interaction. A situation where a user observes a critical situation in the simulated system and dispatches a constrained resource during the execution of the simulation program is an example of user-initiated interaction. Under the simulator view, interactive simulation architectures can be used to develop human-in-the-loop simulators. Such simulators can provide a powerful synthetic environment for training of human operators in complex systems. Interaction with a high-fidelity, synthetic representation of the system can be effective in enhancing user understanding of the complexities of the large, dynamic system.

The major challenges in developing interactive simulations are problems associated with computer hardware (Bell and O'Keefe 1987). Bell (1991) highlights the historic struggle of the early VIS development efforts with advances in computer hardware. Early VIS systems including See–Why were developed for large main frames. During the 1980s and 1990s, personal computers and workstations became the standard for systems development. Most VIS packages available then were hardware dependent and suffered from problems of portability.

With the advent of object-oriented programming and specifically the Java programming language, hardware-independent simulations came to the fore in the late 1990s. An example of a Java-based application is JADIS (Narayanan et al. 1997). JADIS uses the model view controller (MVC) paradigm from Smalltalk (Goldberg 1990). The simulation model allows multiple interfaces, which are separate processes that execute concurrently on different machines. JADIS integrates concepts from object-oriented programming, concurrent processing, and graphical user interfaces (GUIs) to provide a powerful design approach to interactive simulations.

The JADIS architecture facilitates development of interactive simulations. This capability goes well beyond animating discrete-event simulations such as those seen in most commercial packages. In JADIS simulations, users can not only alter the parameters of the simulation, but also modify the system dynamics. For example, users in the airbase logistics simulation can alter the parameters of the maintenance resources and the mode of sortie generation at run time. Real-time human decision making can therefore be readily studied using JADIS simulations. The JADIS architecture was developed on a UNIX workstation under the Solaris operating system. JADIS was successfully executed on several platforms including a personal computer and a Macintosh without altering a single line of the source code.

1.2.1.3 Web-Based Applications

With the advent of the Internet, the interest in applying the World Wide Web infrastructure to simulation modeling and analysis increased (Fishwick 1996). An excellent collection of web-based simulation resources is on the Internet (Page 1998). Several researchers in the simulation field strongly believe that the web is likely to influence several areas of simulation including model building, execution, and sharing (Nance 1998). Examples of java-based simulations including Simkit (Buss and Stork 1996) and Simjava (McNab and Howell 1996) continued to grow. Shen (1998) described a CORBA simulation facility for distributed simulations. Many of these simulations used Java for portability, reuse, object-orientation, Internet programming, and graphical capabilities and could all be run on the web using a Java-enabled web browser. In all these applications, the level of interaction with the human was somewhat limited. We extended the JADIS architecture to enable high levels of user interaction using the web (Narayanan et al. 1998).

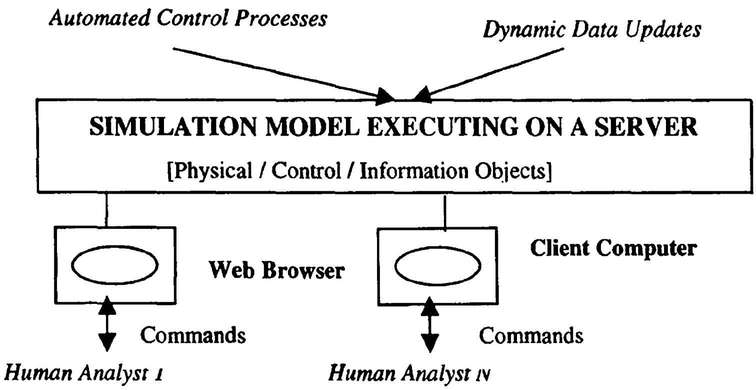


Fig. 1.2 Overall framework for applying the JADIS-Web infrastructure

JADIS-Web (Narayanan et al. 1998) extends the JADIS concepts and implementation on two major features. First, JADIS-Web facilitates the development of multiple views to the same underlying simulation model. Thus, modelers can tailor interfaces to simulations according to the context of user interaction. Second, JADIS-Web supports execution of simulations from multiple platforms with concurrent user interaction from distributed locations. Fig. 1.2 illustrates the overall framework in the application of JADIS-Web infrastructure.

The simulation model of the system being studied executes on a server. The objects in the simulation encapsulate the entities and their interrelationships along a physical/control/information classification, which enables rapid assembly of simulated objects and facilitates what–if analysis. While the simulation is executing, some of the data associated with information storage objects, for example, could be updated through sensors from the real world. Exogenous automated control processes could update the state of the simulated system. An example of this process is an optimization program that computes a problem solution given the current state of the system and the goals of the system and inputs the solution to the simulated system. Several human analysts can connect into the executing simulation model from different locations either through a web browser or from a client computer. They can observe the state of the simulated system and can either modify the simulation parameters such as clock speed or system parameters such as incorporating an additional maintenance resource at run time through a tailored graphical user interface. This type of computational infrastructure is particularly useful in training human operators in a hybrid environment where portions of the system model implemented synthetically are integrated with real- world objects and for analyzing supervisory control issues.

Subsequent application in studying the role of automation and human operator control of remotely piloted vehicles (RPVs) led us to the design and implementation of the UMAST architecture (Narayanan et al. 1999). UMAST is a web-based modeling and simulation architecture for supporting studies of human/system interaction in uninhabited air vehicles. The UMAST architecture supports interactive simulations and can be used to build distributed interactive simulations (DIS) where individual components of the simulation model run on different platforms concurrently. The simulation server in UMAST can be partitioned and executed on distributed platforms. The UMAST architecture integrates concepts such as model-view-controller, distributed computing, web-based simulations, cognitive model-based high-fidelity interfaces, and object-based modeling methods. The architecture supports the modeling requirements for analyzing uninhabited aerial vehicles including interactivity, multi-user connectivity, reconfigurable user interfaces, and modularity of software abstractions in the infrastructure.

1.2.1.4 Personal Digital Assistants

Both JADIS-Web and UMAST were adapted to run on a Palm Pilot. The challenges were more from an implementation perspective and memory limitations associated with the personnel digital assistant (PDA) device. Additionally, the challenge with the user interface design is the footprint limitation with the PDA display screen. Most simulations that currently run on PDAs are in simple problems. With the increasing popularity of iPhones and iPads we anticipate more interactive simulations executing on the operating systems of these PDAs.

1.2.2 Programming Language

From the discussion on the interface evolution, it can be seen that the implementation of interactive simulations has evolved from FORTRAN (e.g., FORSSIGHT), to C, and recently toward object-oriented programming (OOP) languages, e.g., C++, Prof iSEE in Smalltalk-80. One reason for the tremendous appeal of OOP is a major change in abstraction, i.e., moving from the traditional "seize-hold-release" paradigm of simulation languages to a more "natural mapping" paradigm. This is made possible by the object construct, which allows a one-to-one mapping between objects in the system being modeled and their abstractions in the simulation model. Although Java and C++ are both Object-oriented programming languages, C++ is machine dependent, i.e., the software compiled on one platform enables us to run the application on only that platform. JAVA offers an alternative where you can compile the program once and then run the program anywhere without having to compile it, thus making it highly portable. Also using JAVA applets, the simulation can be executed using the web from distributed locations.

In languages such as C++, the software to display the simulation model and to facilitate user interaction are often embedded in the simulation model. Such tight integrations make it difficult to maintain large simulation programs and pose limitations in the development of multiple interfaces to a simulation model.

The coupling of the simulation model with the interface also makes it difficult for the concurrent development of simulation models and their user interface. The advantage that JAVA and C# have over C++ is that they can be used for creating simulations as well as for creating interfaces to simulations.

1.2.3 Programming Paradigm

The evolution of the programming paradigm mirrored the development of programming languages. Numeric programming paradigms were predominantly used in the 1970s. The FORTRAN programming language falls in this category. FORTRAN is a high-level language, i.e., the FORTRAN code looks a lot like English and so complicated programs can be developed very quickly with a minimum level of programming. FORTRAN was designed for engineering application involving numerical analysis and had very little support for visual interfaces or data structures to directly support simulation. These drawbacks paved the way for Functional programming paradigms with data structures in the 1980s. C programming language also allows the developer to manage memory and hardware by the use of data structures and memory addressing. The major disadvantage with C programming language was it was still a mid-level language. C++ made its way in the 1990s. C++ supported Object-Oriented Programming and the features of data encapsulation and software reuse that enabled modular programming and software libraries for rapid development of software applications. The advantages of object-oriented programming for simulation modeling in terms of modularity, software reuse, and natural mapping with real-world entities enabled their application to develop interactive simulations during the 1990s (Narayanan et al. 1997).

In addition to the features of natural mapping between real-world objects and their software abstractions, the power of modularity and the ability of the variants of object-oriented languages (such as Java applets for Java) to the web browser applications has made the object-based paradigm the most popular way of building interactive simulations. Concurrently, with increasing capabilities of networked computers, the power of distributed computing is on the increase. We anticipate distributed interactive applications to emerge even more in the future, particularly because of the popularity of multi-player games in the entertainment industry and war gaming in the defense sector.

1.3 Architectural Issues

In our view, in addition to object-based programming, three major aspects from the computer architecture perspective have had the most impact on interactive simulations development: (a) model-view-controller paradigm, (b) reusable Graphical user interfaces, and (c) distributed computing concepts for networked Computing.

1.3.1 Model View Controller

Model view controller (MVC) is a paradigm for developing graphical user interface software in a modular manner (Goldberg 1990; Gobbetti and Turner 1991; Krasner and Pope 1988). In MVC, any interactive program is conceptually divided into three areas: (1) the *model,* which contains a representation of the application domain, (2) the *view,* which contains the specification of the visual interface, and (3) the *controller,* which contains a specification of the user interactions with the underlying model. In the context of interactive simulations, the model refers to the simulation model, the view corresponds to GUI, and controller refers to the processing of user input to query and modify the simulation model dynamically.

The application of the MVC framework provides several potential advantages to interactive simulations. First, the separation of the model from the view allows model development to take place concurrently with the specification of the interface. The simulation developer can focus on the model development and leave the responsibility of the interface design to a GUI designer. Second, the paradigm supports multiple views into the same simulation. The developer can then plug different displays into the simulation. The simulation model can be reconfigured to meet the needs of individual modelers without requiring developer interaction. Thus, the productivity of software development is enhanced. The reuse of existing designs and refinements potentially also leads to a stable suite of applications with a consistent style.

MVC is an improvement over previous approaches to developing interactive simulations. Simulation modelers no longer need to be experts at implementing simulation models as well as designing and implementing graphical displays. GUI experts can create graphical libraries for the simulation analysts to customize the simulation view. With MVC, many users can access multiple, simultaneous views of the same simulation model. Many interactive simulations from the mid-1980s have applied this framework for software systems development.

1.3.2 Reusable Graphical User Interfaces

The type and quality of the visual display is extremely important for a Human-in-the-loop simulation. Gerlach and Kuo (1991) found that a faulty interface can potentially trap the user in undesired situations, therefore affecting users' attitudes towards the application. An abstract interface also limits the human to fully comprehend the meaning of the results of the simulation. Cohen (1991) indicated that most of the simulations that are suited for experts are not easy to interpret by a novice user. The advent of object-based programming and the growing interest in making systems more user friendly led to enhancements in direct manipulation widgets and graphical interfaces that were generated from software libraries through the object-oriented programming constructs. For C++ based simulations on UNIX platforms, Motif widgets were popular. With the growth in Java and its variants, support for graphical user interface development came concurrently with the core language support.

1.3.3 Distributed Computing Concepts

Growing interest in the US Department of Defense (DoD) in interactive simulations from a war gaming perspective led to creation of standards such as DIS architectures and high level architecture (HLA). Commercially, there were major developments in addressing legacy systems integration through middleware software such as CORBA. These developments also found some applications in interactive simulations where there was always an interest in displaying user interfaces at high-fidelity, particularly in real-time applications with minimal latency. Interactive simulation developers were interested in decomposing simulation code and user interfaces in multiple computer platforms to harness the computing power in networked platforms for real-time rendering of high-fidelity graphics. For web-based interactive simulation applications, distributed computing concepts enabled the maintenance of system state in a central server and enabled synchronized display of the system state on web browsers that remained agnostic to the specific system state at the user interaction level (Narayanan et al. 1998).

1.4 Overview of Application Trends

Beyond the early applications for what-if analyses, most interactive simulations primarily focused on simulator-based training. Subsequent applications have examined the role of humans in semi-autonomous systems control (Narayanan et al. 1999), flexible manufacturing systems scheduling (Dunkler et al. 1988), and knowledge acquisition associated with human decision making (Craig et al. 2001). Most recently, there has been interest in the development of constructive simulations where the focus is on developing automated agents that represent human decision making in the software abstractions. These approaches can potentially work in applications where the problem solving behavior of the human is completely understood. Human reasoning in many complex, dynamic systems is not fully understood due to the tacit dimension that embodies expertise (Polanyi 2009), especially involving team problem solving. Hence, human-in-the-loop simulations will continue to serve as irreplaceable tools in the foreseeable future for complex systems analysis!

References

Bell PC (1991) Visual interactive modeling: the past, the present, and the prospects. Eur J Oper Res 54:274–286

Bell PC, O'Keefe RM (1987) Visual interactive simulation—History, recent developments, and major issues. Simulation 49(3):109–116

Bishop JL, Balci O (1990) General purpose visual simulation system. In: Proceedings of the 1990 winter simulation conference, pp 504–512

Buss AH, Stork KA (1996). Discrete-event simulation on the world wide web using Java. In: Proceedings of the 1996 winter simulation conference, San Diego, CA, 780–785

Cohen PR (1991) integrated interfaces for decision-support with simulation. In: IEEE computer society proceedings of the 23rd conference on winter simulation

Conway R et al (1995) User's Guide to Xcell Plus: Factory Modeling System, Release 4.0, Scientific Press, Palo Alto, California

Craig K, Todd J, Giest J, Narayanan S, Hoagland D (2001) Predicting human operator actions for a constructive simulation using genetic algorithms. In: Proceedings of the 2001 society for computer simulations summer simulation conference

Dunkler O, Mitchell CM, Govindraj T, Ammons JC (1988) The effectiveness of supervisory control strategies in scheduling flexible manufacturing systems. IEEE Trans Sys Man Cybern 18:223–237

Fishwick PA (1996) Web-based simulation: some personal observations. In: Proceedings of the 1996 winter simulation conference, San Diego, CA, 772–779

Gerlach JH, Kuo FY (1991) Understanding human–computer interaction for information systems design. MIS Q 15(4):527–549

Gobbertti E, Turner R (1991) Object-oriented design of dynamic graphics applications. In: New trends in animation and visualization. John Wiley and Sons, Chichester, UK, pp 43–58

Goldberg A (1990) Information models, views, and controllers. Dr. Dobb's J 54–61

Hurrion RD (1980) An interactive visual simulation system for industrial management. Eur J Oper Res 5(2):86–94

Hurrion S (1978) Visual interactive simulation: an aid to decision making. Omega 6(5):419–426

Kirkpatrick PF, Bell PC (1989) Visual interactive modeling in industry: results from a survey of visual interactive model builders. Interfaces 19(5):71–79

Krasner GE, Pope ST (1988) A cookbook for using the model-view controller user interface paradigm in Smalltalk. J Object Oriented Program – JOOP 1(3):26–49

Lyu J, Gunasekharan (1993) Developing a visual interactive simulation model for flexible manufacturing systems. Int J Oper Prod Manag 13(6):59–67

McNab R, Howell FW (1996) Using java for discrete event simulation. In: Proceedings of the twelfth UK computer and telecommunications performance engineering workshop (UKPEW), University of Edinburgh, 219–228

Musselman KJ (1986) TESS as a catalyst for change. TIMS/ORSA Joint National Conference

Nance RE (1998) Simulation modeling methodology in the wonderfully webbed world. In: Proceedings of the 1998 international conference on web-based modeling and simulation. Available on: http://www.isima.fr/scs/wbms/d36/nanceRE.html

Narayanan S, Schneider NL, Patel C, Reddy N, Carrico TM, Dipasquale J (1997) An object-based architecture for developing interactive simulations using Java. Simulation 69(3):153–171

Narayanan S, Malu P, Krishna APB, Dipasquale J, Carrico TM (1998) A web-based interactive simulation architecture for airbase logistics system analysis. Int J Ind Eng, 5(4):324–335

Narayanan S, Edala NR, Geist J, Kumar K, Ruff HA, Draper M, Haas M (1999) UMAST: a web-based architecture for modeling future uninhabited aerial vehicles. Simulation 73(1):29–39

O'Keefe RM (1987) What is visual interactive simulation? (and is the methodology for doing it right?). In: IEEE Proceedings of the 1987 winter simulation conference, pp 461–464

Page E (1998) A survey of web based simulations. Available on: http://ms.ie.org/websim/survey/survey.html

Paul RJ (1989) Visual simulation: seeing is believing. In: Sharda R, Golden BL, Wasil E, Balci O, Stewart W (eds) Impacts Recent Computer Advances on operations research, publications in operations research Series, vol 9. North-Holland, New York, pp 422–432

Pedgen LA et al (1985) Graphical interpretation of output illustrated by a SIMAN manufacturing simulation model. In: Proceedings of the 1985 winter simulation conference, pp 244–251

Polanyi M (1966) The tacit dimension. The University of Chicago Press. Routledge, London (ISBN 978-0-226-67298-4 2009 reprint)

Shen CC (1998)" Discrete-event simulation on the internet and the Web." In: Proceedings of the 1998 international conference on web-based modeling and simulation. Available at: http://www.isima.fr/scs/wbms/d31/Web_sim.html

Chapter 2
Performance Measurement and Evaluation in Human-in-the-Loop Simulations

Ling Rothrock

Abstract A prerequisite for designers of complex systems is a proper understanding of human performance characteristics. While human factors texts provide some insights into basic performance issues, the emergence of highly-automated computing systems have fundamentally altered the way humans work. The purpose of this paper is to present an approach to quantify and analyze human performance in human-in-the-loop simulations based on over ten years of research experience. The approach is centered on a measurement construct, called a time window, which enables a functional relationship between constraints on operator activities and time availability. A blackboard model is presented as the mechanism to generate, maintain, and complete time windows. To demonstrate the utility of time windows, an existing implementation in a real-time human-in-the-loop simulation is also described. An extension of time windows to measure team performance is also discussed. Using time window outcomes, samples of previous analyses are presented to exhibit the potential of the construct.

2.1 Foundations

2.1.1 Introduction

The emergence of highly-automated computing systems has fundamentally altered the way humans work. As these systems have increasingly become mediators between human operators and the work environment, human understanding of how

L. Rothrock (✉)
The Harold and Inge Marcus Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University, Leonhard Building 210, State College,
University Park, PA 16802, USA
e-mail: lxr28@engr.psu.edu

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_2,

work is accomplished has greatly diminished. Remarks of "What happened?" or "Why did it do that?" are not uncommon as operators seek to understand the processes of systems designed to improve their work. Rather than serving the purpose of being tools for human use, these systems have come to be regarded as autonomous agents to which humans must adapt in the workplace.

To investigate human decision making in these highly-automated systems, researchers have had to rethink the applicability of traditional laboratory methods such as expected utility theory (Beach and Lipshitz 1993). The use of traditional methods assumed that findings from the laboratory environment—where highly cognitive, single-choice tasks were conducted—could be applied to more realistic settings. The premise that findings from a static, forced-choice task can be extended to an operational environment has been called into question (Hammond 1986). In fact, some researchers have recommended that studies of human operators must occur in settings that are representative of the actual environment (Suchman 1987; Endsley 2006).

The purpose of this paper is to present a research approach to quantify and analyze human performance within a human-in-the-loop simulation based on over ten years of research experience. The key concept introduced here is the notion of a time window that provides a functional relationship between constraints on operator activities and time availability. A methodology is proposed to evaluate time windows as well as to assess operator attunement to them. This paper contains reprints of three journal articles. Section 2.1 is based on Rothrock (2001) that explains the foundations of time windows, Sect. 2.2 is based on Rothrock et al. (2009) that explores team performance measurement, and Sect. 2.3 is based on Ma et al. (2011) which extends performance measurement to service-based industries like call centers.

2.1.2 Situations, Constraints and Time Windows

2.1.2.1 Situativity Theory

In order to extract situations, constraints, and available time, these terms must first be clearly defined. The meaning of the terms "situation" and "constraint" as they have been used thus far is consistent with the interpretation provided by Greeno and Moore (1993) and Greeno (1998). They introduced a theory of situativity in which cognitive processes are analyzed as relations between operators and other subsystems in the environment. The theory is powerful because it stipulates that a functional relationship exists between an operator's decision making activities and the task environment. The dependency relation between an action and the resultant situation—also known as a constraint—contains the following form (Greeno 1994, p. 339):

$$\ll\texttt{action by operator}\gg \texttt{=>} \ll\texttt{good effect in situation}\gg,$$

where the good effects are outcomes that are required for a broader activity to be successful.

2.1.2.2 Time Window Extension to Situativity Theory

The notion of a time window is an extension to situativity theory. To computationally implement the time window extension, therefore, a greater degree of definitional precision is required. Accordingly, the definition of time windows conveys the concepts of situativity theory while relying upon temporal logic (Gabbay et al. 1994; Allen 1983) to provide the basic foundation for a computational model.

A time window is a construct that specifies a functional relationship between a required situation and a time interval that specifies availability for action. A time window does not specify what action must be taken, but only that there exists an action which will result in the required situation. In the course of operator activity within a dynamic task, n time windows are denoted as w_i for $i = 1$–n.

At the onset of operator interaction, all time windows are designated as inactive and represented by the set U_0. Until a time window is designated as open, it remains inactive. Time windows are designated as open if the availability for action exists for a required situation at the current point in time space. The set of open time windows at time t is designated as O_t. When a required situation no longer exists, the corresponding time window is designated as closed. The set of closed time windows at time t is denoted as C_t. The membership of U, O, and C is defined to be persistent over time, and will remain the same (i.e., $U_{\mathrm{t+1}} = U_{\mathrm{t}}$, $O_{\mathrm{t+1}} = O_{\mathrm{t}}$, and $C_{\mathrm{t+1}} = C_{\mathrm{t}}$) unless designated otherwise. Methods to extract conditions specifying the opening and closing of time windows will be covered in Sect. 2.1.3.

To complete the constraint specified by situativity theory in a temporal context, one must define operator action and the relationship between action and time window. An operator action is defined here as a two-tuple that includes a detectable act performed by the operator at a specific point in time. In the course of operator interaction within a dynamic task environment, m actions are denoted as $\mathbf{b}_{\mathrm{j}}$ for $j = 1$ to m. The relationship between action and time window can be described by two Boolean indicator functions, I_w^l, such that, for l = 1, the function evaluates whether an action meets the required situation specified by a time window, and for l = 2, the function evaluates the relevance of an action toward a time window.

Thus,

$$I_w^1(\mathbf{b}) = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{b} \text{ meets situation specified in } \boldsymbol{w} \\ 0 & \text{if } \mathbf{b} \text{ does not meet situation} \end{array} \right\}, \text{ and}$$

$$I_w^2(\mathbf{b}) = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{b} \text{ is relevant toward } \boldsymbol{w} \\ 0 & \text{if } \mathbf{b} \text{ is not relevant toward } \boldsymbol{w} \end{array} \right\}.$$

Six predicates, $M_T^k(w_i, \mathbf{b}_j)$ for $k = 1$–6, will now be constructed to characterize fundamental relationships between time windows and operators actions over a time interval T. In particular, the truth value, $||M^k(w_i, \mathbf{b}_j)||_{T+,T-}$, of each predicate is evaluated for a time interval that starts when operator interaction in the task begins (T+) and ends when operator interaction ceases (T−). Given that $\mathbf{b}_j$ occurs at time s, equations to evaluate the first five predicates are listed as follows:

- An on-time action that results in a required situation, $M_T^1(w_i, \mathbf{b}_j)$, is formally defined as,

$$||M^1(w_i, \mathbf{b}_j)||_{T+,T-} = 1 \text{ iff } \exists i \text{ such that } [I^1_{w_i}(\mathbf{b}_j) = 1] \wedge (w_i \in O_s); \quad (2.1)$$

- An early action that results in a required situation, $M_T^2(w_i, \mathbf{b}_j)$, is defined as,

$$||M^2(w_i, \mathbf{b}_j)||_{T+,T-} = 1 \text{ iff } \exists i \text{ such that } [I^1_{w_i}(\mathbf{b}_j) = 1] \wedge (w_i \in U_s); \quad (2.2)$$

- A late action that results in a required situation, $M_T^3(w_i, \mathbf{b}_j)$, is defined as,

$$||M^3(w_i, \mathbf{b}_j)||_{T+,T-} = 1 \text{ iff } \exists i \text{ such that } [I^1_{w_i}(\mathbf{b}_j) = 1] \wedge (w_i \in C_s); \quad (2.3)$$

- An action that is relevant toward a required situation, but does not result in it, $M_T^4(w_i, \mathbf{b}_j)$, is defined as,

$$||M^4(w_i, \mathbf{b}_j)||_{T+,T-} = 1 \text{ iff } \exists i \text{ such that } [I^1_{w_i}(\mathbf{b}_j) = 0] \wedge [I^2_{w_i}(\mathbf{b}_j) = 1]; \quad (2.4)$$

- An action with no corresponding time window, $M_T^5(\mathbf{b}_j)$, is defined as,

$$||M^5(\mathbf{b}_j)||_{T+,T-} = 1 \text{ iff } \forall i, \ (I^2_{w_i}(\mathbf{b}_j) = 0). \quad (2.5)$$

Because the sixth predicate is based on a time window instead of action, the equation to evaluate it is defined separately as follows:

- A time window that has been missed, $M_T^6(w_i)$, is defined as,

$$||M^6(w_i)||_{T+,T-} = 1 \text{ iff } \forall j, \ (I^2_{w_i}(\mathbf{b}_j) = 0). \quad (2.6)$$

Because of their reliance on temporal logic, Eqs. 2.1–2.5 offer a more explicit description of constraints than the conceptual distinctions offered by situativity theory. Specifically, the time window framework can now be utilized as a dependency relation between an action and a required situation that is also bound by time.

2.1.2.3 Extracting Time Window Information

To extract time window information, one must view operator decision making in its experiential context. The focus of the extraction is, therefore, on the use of analysis methods to discover mappings between operator actions and situations required to meet system objectives.

Three techniques meet the criteria for extracting time window information. Because each technique focuses on a slightly different information source, the most effective approach is one that integrates the advantages of all three. One method, cognitive task analysis (CTA) (e.g., Militello and Hutton 1998), is based on human input. CTA focuses on experienced practitioners in operational contexts to extract information they deem diagnostic to successfully operate in the task environment. The two other methods rely on theoretical and empirical studies of the environment in which the task is performed. Cognitive work analysis (CWA) utilizes theoretical expertise and engineering analyses of system dynamics to identify conceptual distinctions within a work domain that can later be used as modeling tools (Vicente and Rasmussen 1992). Ecological task analysis (ETA) is focused on analysis of the task environment to determine empirical regularities in environmental behavior (Kirlik 1995). Time window information extracted through the integrated method should therefore be: valid from an operator's perspective; consistent with system dynamics; and true to the availability of action within the task environment. Consider, for example, the process of extracting time window information in an air traffic control (ATC) domain. CTA is used to determine normal operator courses of actions to reach established objectives. CWA is used to ascertain static and kinematic constraints in the ATC domain that affect the operator's ability to reach the objectives (e.g., radar range). ETA is used to discover constraints in the ATC environment (e.g., appropriate regulations) and empirical regularities to which good controllers must be sensitive.

Once time window information has been extracted, the next step in the proposed research methodology is to implement the construct. The next section presents an object-oriented simulation architecture that includes a time window generation and maintenance system based on the blackboard model.

2.1.3 Blackboard Model in Object-Oriented Simulations

The blackboard model was first developed in the early 1970s as a tool for speech understanding (Erman et al. 1980). Since then, it has been implemented in many domains for multiple purposes. For example, Vranes et al. (1991) have used it as a tool to conduct military planning. Rubin et al. (1988) used it as a framework to construct an operator's associate in a supervisory control task. More recently, Adeli and Yu (1995) used it to develop an integrated computing environment to solve complex engineering problems. Although it has been implemented in vastly different forms, the blackboard model approach to problem solving remains the same. In essence, the blackboard model of problem solving is a reasoning scheme which applies pieces of knowledge at the most opportune time to construct a solution to the problem.

A blackboard model consists of three major components (Nii 1986): knowledge sources; the blackboard data structure; and control. The knowledge sources contain knowledge required to solve the problem. The blackboard data structure is a global

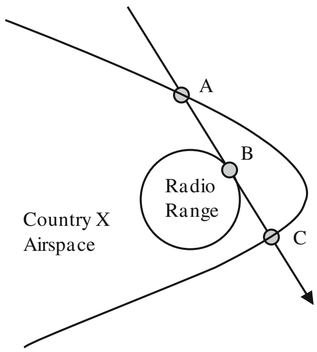


Fig. 2.1 Air traffic control example. An unknown aircraft enters Country X airspace at point *A*, enters and leaves range to establish radio contact at point *B*, and leaves Country X airspace at Point *C*

database in which partial and full solutions are kept. The blackboard control is an opportunistic reasoning model that guides problem solving by choosing and activating appropriate knowledge sources.

2.1.3.1 The Blackboard and Time Windows

To illustrate the use of blackboard model to open, maintain, and close time windows, consider the following example: In a real-time simulation, a human operator assumes the role of an ATC monitoring aircraft entering and leaving Country X's airspace (Fig. 2.1). The operator has been given specific instructions to identify all unknown aircraft entering the airspace, and to establish radio contact with all aircraft that come within radio range. An unknown aircraft, traveling along the trajectory indicated by the direction vector, enters Country X airspace at point A, enters and leaves range to establish radio contact at point B, and leaves Country X airspace at point C.

In the context of time windows, the blackboard knowledge sources include operators who act on the environment, and entities that produce situations. These sources contribute not only actions and situations to the blackboard, but also temporal information that defines constraints within the environment in which the task is performed.

In the example, the knowledge sources include the ATC and the unknown aircraft. Moreover, the unknown aircraft also reveals constraints that dictate expected ATC actions. At point A, w_1 is designated as open so that $w_1 \in O_{t_a}$ with the specification that the situation of a correctly identified aircraft be required. The time at which the aircraft reaches point A is designated as t_a. At point B, a second time window, w_2, is designated as open to specify the situation of established radio contact at time t_b so that $w_2 \in O_{t_b}$. Since the trajectory of the aircraft is tangential to the curve bounding the radio contact area, the available time interval for the ATC to establish radio contact is instantaneous. Therefore, w_2 is also designated as closed at time t_b so that $w_2 \in C_{t_b}$. At point C, the aircraft exits Country X airspace and triggers the closing of w_1 so that $w_1 \in C_{t_c}$.

The blackboard data structure holds time window information in the form of computational and solution-state data. Each time window represents a structural means-ends hierarchy (Vicente and Rasmussen 1992) where the required situation (ends) is achieved by an expected operator action (means).

While the knowledge sources provide necessary information to generate and maintain time windows within the blackboard architecture, the activities on the blackboard are monitored and controlled by the blackboard control. The control uses opportunistic reasoning to apply backward reasoning as well as forward reasoning models to maintain time window information. Backward reasoning is applied at the point of a required situation to determine if the expected operator action has been taken, while forward reasoning starts at an operator action to determine if the action outcome meets any required situations.

Returning to the ATC example, assume that the operator takes three actions. The first action, $\mathbf{b}_1$, incorrectly identifies the aircraft at time t_1, where t_1 is before t_a (i.e., $t_1 < t_a$). The second action, $\mathbf{b}_2$, correctly identifies the aircraft at time t_2 where $t_a < t_2 < t_c$. The third action, $\mathbf{b}_3$, alerts Country X's border patrol at time t_3 where $t_b < t_3 < t_c$.

Using backward reasoning, the blackboard control examines all open time windows to determine if any has been met. At time t_a, the control assesses $\mathbf{b}_1$ as applicable to w_1 so that $I^2_{w_1}(\mathbf{b}_1) = 1$, but does not result in the required situation so that $I^1_{w_1}(\mathbf{b}_1) = 0$. Thus, Eq. 2.4 is satisfied and the action is deemed irrelevant. At time t_2, the control determines that $\mathbf{b}_2$ is consistent with the expected operator action specified by w_1 so that $I^1_{w_1}(\mathbf{b}_2) = 1$. Moreover, because $w_1 \in O_{t_2}$, the control evaluates w_1 and $\mathbf{b}_2$ to satisfy Eq. 2.1 and assesses $\mathbf{b}_2$ an on-time, required action.

Applying forward reasoning, the control examines all current actions to determine if they address any required situations. At time t_3, the control determines that $\mathbf{b}_3$ is not relevant toward any time window so that $\forall i,\ I^2_{w_i}(\mathbf{b}_3) = 0$. The control does not, however, make a determination on the action at this point. Rather, it seeks resolution of the action's status by checking backward reasoning results to ensure that the action is not early for a later time window. Nevertheless, the third action was eventually determined to be irrelevant.

2.1.3.2 Blackboard Models in a Real-Time, Object-Oriented Simulation

Conceptually, the use of time windows in a blackboard model has been demonstrated. To illustrate the utility of time windows in a simulation environment, the implementation of time windows via a blackboard model will now be presented. The simulation architecture developed at the Georgia Institute of Technology (Chu et al. 1991; Jones et al. 1995) is used as a baseline for discussion. The integration of the blackboard model within the simulation architecture is depicted in Fig. 2.2.

The active simulation object (ASO) is used as a base class so that events can be scheduled by methods contained in its subclasses. The display class contains

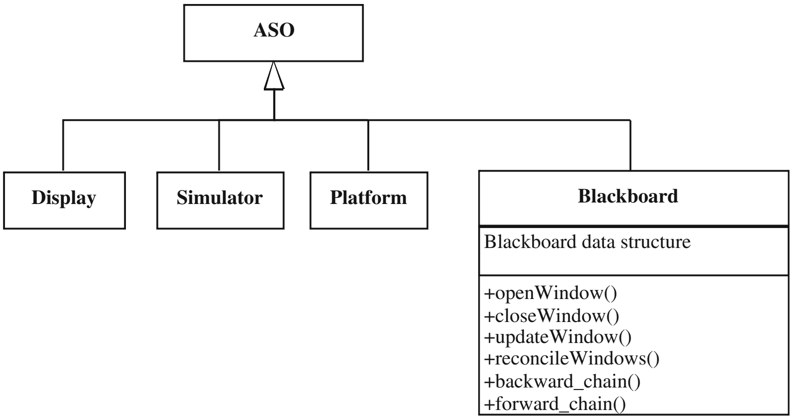


Fig. 2.2 Simulation class diagram

parameters as well as methods for generating the graphical user interface. The simulator class contains methods to control the experimental simulation. The platform class represents physical platforms (e.g., airplanes) that exist in the simulation environment, and contains methods that allow those objects to act upon the environment. The blackboard class contains the knowledge sources within the blackboard data structures. It also contains methods to control the blackboard by opening time windows, closing time windows, updating and reconciling time windows, conduct forward chaining reasoning, or execute backward chaining reasoning.

An illustration of the blackboard and time window implementation within an object-oriented simulation framework is represented in the form of a sequence diagram in Fig. 2.3. A sequence diagram is a model that describes how groups of objects collaborate in some behavior (Booch et al. 1999). Each box above the diagram represents an object. Each vertical line represents the object's life during the interaction. The flow of events is chronologically ordered from top to bottom. Methods labeled with an asterisk are iterative.

Revisiting the air traffic control example, the event flow of operator actions and aircraft movements is reflected in Fig. 2.3. A chronologically-ordered narration on the sequence of events follows:

1. The flight of the unknown aircraft along the southeasterly trajectory is accomplished by the iterative call of the modifyPosition() method.
2. The first operator action, $\mathbf{b}_1$, of incorrectly identifying the aircraft (as a jet) is posted to the blackboard.
3. When the control detects the aircraft entering the airspace of Country X, w_1 is designated as open.
4. The backward-chaining model reasons that $\mathbf{b}_1$ is an incorrect identification that has been taken early. Thus, $||M^4(w_1, \mathbf{b}_1)||_{T+,T-} = 1$.

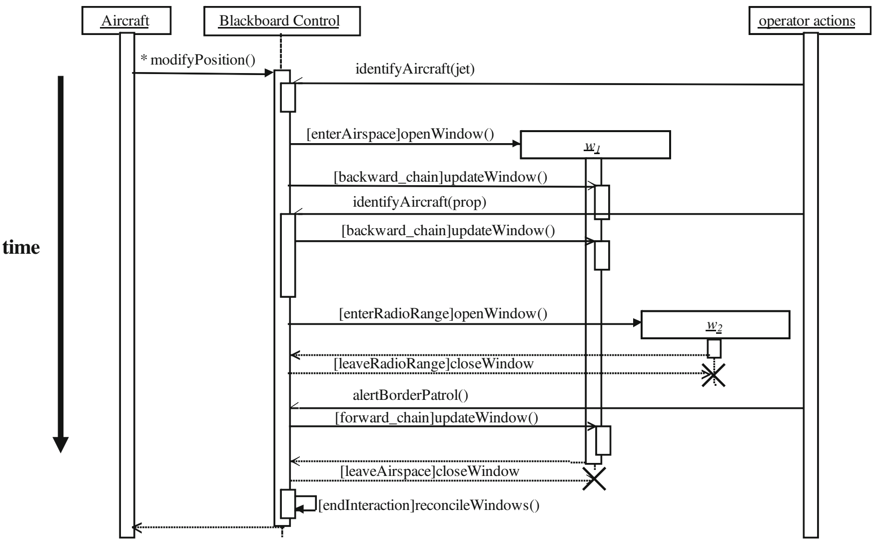


Fig. 2.3 Time window sequence diagram

5. The second operator action, $\mathbf{b}_2$, of correctly identifying the aircraft (as a propeller-driven aircraft) is posted to the blackboard.
6. The backward-chaining model determines that a correct identification action has been taken on-time. Therefore, $||M^1(w_1, \mathbf{b}_2)||_{T+,T-} = 1$.
7. When the control detects the aircraft entering radio range, w_2 is designated as open.
8. The control immediately detects the aircraft leaving radio range, and closes w_2.
9. The third operator action to alert the border patrol, $\mathbf{b}_3$, is posted to the blackboard. The forward-chaining model determines that no time window specifies the need for $\mathbf{b}_3$. Moreover, the action does not serve any required situation—radio contact or correctly identified aircraft. Therefore, the action is classified as irrelevant. Thus, $||M^5(\mathbf{b}_3)||_{T+,T-} = 1$.
10. When the control detects the aircraft leaving Country X airspace, w_1 is closed.
11. Operator interaction ceases as the aircraft leaves Country X airspace. At this point, the backward-chaining model reconciles the blackboard by closing all open windows and assessing if windows have been missed. The only window in question is w_2, and is assessed to be missed so that $||M^6(w_2)||_{T+,T-} = 1$.

2.1.3.3 Possible Time Window Outcomes

The utility of a time window is not only in its temporal and functional descriptions, but also in the richness of the possible outcomes. Some time window outcomes have already been described. Not surprisingly, the complete space of possible time

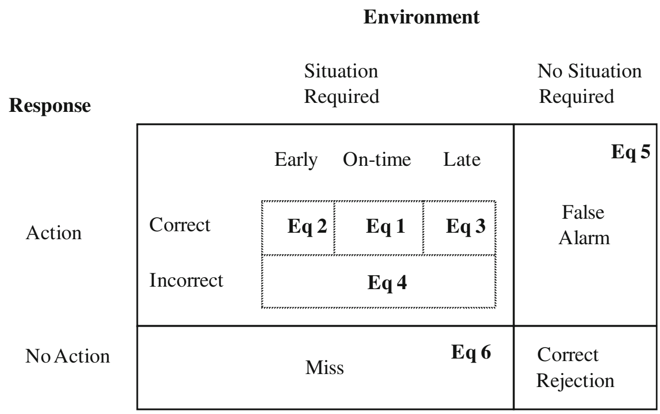


Fig. 2.4 Possible time window outcomes. The environment is delineated in terms of situation required (time window exists) or no situation is required (time window does not exist). Equations 2.1–2.4 represent actions that are relevant to a time window. Equations 2.1–2.3 represent actions that result in the required situation (correct actions). Equation 2.4 represents actions that do not meet the required situation (incorrect actions) even though they are relevant

window outcomes (Fig. 2.4) is represented by the fundamental relationships between time windows and operator actions outlined in Eqs. 2.1–2.6. In itself, the existence of a required situation does not impact system performance. It is the presence of operator action in a temporal context that specifies whether performance is good or poor. An incorrect, early action (first ATC operator action) is represented as Eq. 2.4. An on-time, accurate action (second ATC operator action) is represented as Eq. 2.1. An action with no corresponding required situation (third ATC operator action) is categorized as Eq. 2.5. A non-action for an existing situation requirement (no attempt to establish radio contact) is characterized as a miss and is represented as Eq. 2.6.

It has been shown that time window is a viable construct, both conceptually as well as in an implemented mechanism within a simulation framework. However, the value of implementing time windows in a research effort has yet to be discussed. The following section will discuss the implications of applying time windows toward human performance measurement and evaluation.

2.1.4 Time Windows and Human Performance

2.1.4.1 Implications Toward Measurement

Wickens and Holland (2000) observed that most performance measures are associated with one of the following categories of raw data:

1. Measure of speed or time (e.g., how fast can an operator reach for a lever?);
2. Measure of accuracy or error (e.g., how many typing mistakes are made?);

3. Measure of workload or capacity demands (e.g., how difficult is this task?); and
4. Measure of preference (e.g., is one display preferred over another?).

In most cases, the use of a particular type of measure is dependent on the real-world task to which the results of the laboratory task generalize. The emphasis, therefore, is on finding methods that analyze factors in isolation. However, it has already been noted that research on dynamic and complex environments should take place in representative settings. Recognizing the problem, researchers have sought to develop techniques to measure performance in tasks that are more representative of the operational environment. Sanderson et al. (1989) focused on the use of verbal protocol data in operational tasks. Howie and Vicente (1998) used automated log files to construct a number of measures to assess operator performance in a microworld setting. Still other researchers (Raby and Wickens 1994; Moray et al. 1991; Laudeman and Palmer 1995) focused on recorded data in time-critical task environment.

The time window construct represents a fundamental shift from existing performance measurement approaches. It is not focused solely on whether a certain task is completed, or how fast a certain button is pushed, or what percentage of error is detected. Rather, it provides a computational framework to dynamically evaluate heterogeneous situation demands and operator abilities to meet them in a complex domain. The benefit of the framework is the functional link between operator actions and the domain with which she/he interacts.

2.1.4.2 Implications Toward Evaluation

As shown in Fig. 2.5, utilization of the time window construct leads to a multi-dimensional space of possible outcomes. As yet, no mathematical formalism exists to comprehensively evaluate operator performance based on all dimensions. Instead, two methods are proposed to provide different perspectives on operator attunement to the constraints. The first method, factor analysis, is designed to determine correlations among different types of time windows and time window outcomes. The second method depends on the use of signal detection theory (SDT) to determine the sensitivity of operator actions to situation requirements.

Factor analysis is a data reduction technique that attempts to find a smaller number of dimensions, or factors, while retaining most of the information in the original space (Green 1978). The intent, therefore, is to evaluate which situations and operator actions can be aggregated into higher order factors. The analysis process proceeds in three major steps:

1. Rotate original data (i.e., variables consisting of the different time window outcomes in different types of required situations) to a new orientation that exhibits dimensions with maximal variance;
2. Reduce the dimension of the transformed data space; and
3. Identify the new dimensions, or factors, in terms of variables that show high association with each factor.

Fig. 2.5 Signal detection theory outcomes

Response	**State of the world**: Signal	Noise
Detected	Hit	False Alarm
Not Detected	Miss	Correct Rejection

The reader is referred to any multivariate statistics text for details on steps 1 and 2. To identify underlying factors, a technique called the scree test (Cattell 1966) is suggested. In essence, the scree test requires plotting the variance accounted by each factor extracted, and then finding elbow in the curve of the plot. To identify which variables belong to the selected factors, factor loadings (i.e., correlation between the variable with a factor) are recommended.

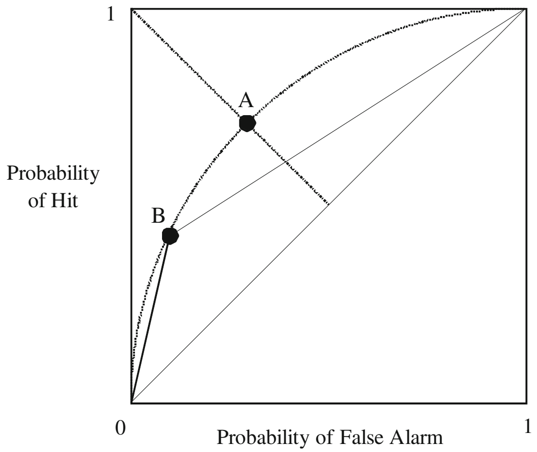
Signal detection theory is a formulation that has been widely used to assess human ability to detect signals (Green and Swets 1966). The premise of the paradigm is that there are two states of the world (signal vs. noise) and two possible human responses (I detect a signal vs. I do not detect a signal). The possible resulting states produces a 2×2 stimulus–response matrix shown in Fig. 2.5.

A key theoretical representation of signal detection theory is the receiver operating characteristic (ROC) (Swets 1996). The standard graphical depiction of the ROC is known as the ROC curve (Fig. 2.6). The curve reveals two important sources of information about operator performance: an individual's decision criterion (the amount of evidence required to detect a signal); and the sensitivity of an individual's detection performance (the individual's ability to discriminate between signal and noise).

In order to apply SDT to the sensitivity analysis of time window outcomes, one must develop methods that do not violate assumptions of either formulation. In particular, the following three issues must be addressed: conversion of time window outcomes to SDT outcomes, calculating the probability of a false alarm in time window outcomes, and the development of a sensitivity measure without distribution assumptions.

The conversion of time window outcomes (Fig. 2.4) to SDT outcomes is dependent on a common definition of a hit. If a hit is defined to be an on-time and accurate action, so that Eq. 2.1 holds, then conversions from time window outcomes to SDT outcomes can readily be made. Table 2.1 shows the conversion

Fig. 2.6 The ROC curve under different distribution assumptions. If the distributions of signal and noise are normal, the sensitivity, d′, is determined by the distance of a point on the curve, point *A*, from the upper left diagonal. If no assumptions on the distributions can be made, the sensitivity can be approximated by the area under ROC (e.g., point *B*)



from time window outcomes to SDT outcomes. If an action is not executed on time, it is considered a false alarm. Therefore, a signal is only considered valid and detectable during a specified time interval in which the associated time window is designated open.

The original SDT formulation required forced-choice tasks primarily to ensure that correct rejections were accurate assessments of the absence of a signal. However, the decision environments for which time windows are intended are dynamic and interactive, and operators are not forced to take action. To calculate the probability of false alarm, which requires the number of false alarms and correct rejections, an accurate accounting method for correct rejections is needed. In fact, one method to measure correct rejections in these "free response" (Wickens and Kessel 1979) environments has already been developed. Wickens and Kessel (1979) computed the probability of false alarms as the number of false alarms divided by the number of false-alarm intervals. In their formulation, equal-valued intervals that span the detection task are separated into those that contain hits, and those that do not—called false-alarm intervals. Based on this concept, a false-alarm interval can be defined in the time window context. Consider the duration of a time window, T, over the lifetime of a simulation, T_s. The number of false-alarm intervals (FAI) can simply be formulated as:

$$FAI = \frac{T_s}{T} - 1 \tag{2.7}$$

The third issue to be addressed is the need for an appropriate sensitivity measure. If the distributions of the signal and noise are normal, the determination of the sensitivity, d', can be visually determined from the ROC curve. In Fig. 2.6, for instance, the closer point A is from the upper left corner, the higher the sensitivity value. However, no assumptions can be readily made about distributions of signal and noise in dynamic domains. Therefore, one must rely on

Table 2.1 Conversion between time window outcomes and SDT outcomes

Time window outcome	SDT outcome
$\lVert M^1(w_i, \mathbf{b}_j)\rVert_{T+,T-} = 1$	Hit
$\lVert M^2(w_i, \mathbf{b}_j)\rVert_{T+,T-} = 1$	False alarm
$\lVert M^3(w_i, \mathbf{b}_j)\rVert_{T+,T-} = 1$	
$\lVert M^4(w_i, \mathbf{b}_j)\rVert_{T+,T-} = 1$	
$\lVert M^5(\mathbf{b}_j)\rVert_{T+,T-} = 1$	
$\lVert M^6(w_i)\rVert_{T+,T-} = 1$	Miss
Correct rejection	Correct rejection

nonparametric measures of sensitivity. Wickens and Hollands (2000) recommend a simple measure based on area under a ROC. The measure, first considered by Green and Swets (1966), is formulated as follows:

$$A_G = \frac{P(H) + [1 - P(FA)]}{2} \tag{2.8}$$

If only one point is acquired on the ROC, such as point B in Fig. 2.6, a sensitivity value can now be calculated. While these measures are still dependent on distributional assumptions (Caldeira 1980), they nevertheless serve as a good first approximation (Craig 1979).

The research methodology proposed here was implemented in a study to investigate tactical decision making performance under stress. For experiment details, see Rothrock (2001).

2.2 Analyses of Team Performance in a Dynamic Task Environment

In this part of the paper, team performance will be assessed from the perspective of time windows. Teamwork, a central component of team research, is not readily observable and must be inferred from the manner in which teams operate. Of particular interest is the measurement and evaluation of teamwork. The goal of this section is to explore the assessment of team data using a temporal accuracy measure called the relative accuracy index (RAI). The generalized mixed model will be used for the statistical analysis because of the type of data (binomial) and of the correlation structure within team members. The statistical procedure is described in detail to guide researchers who encounter similar problems. Using our statistical analysis, we found that participants whose training focused on coordination activities outperformed those whose training did not. Moreover, we found that workload stress accentuates the difference.

2.2.1 Introduction

An understanding of the relationship between team processes, outcomes and performance is a necessary prerequisite to the development of team training processes. Marks et al. (2002) argue that teamwork or team processes are the mediating links that link the relationship between team training and corresponding team outcome (performance) within the setting of input-process-outcome models. Coovert et al. (1990) suggest that team processes relate to the activities, strategies, responses, and behaviors employed in task accomplishment within teams. Team outcomes on the other hand pertain to the outcome of the various team processes. Any team performance measure or TPM (Cannon-Bowers and Salas 1997) must address the process as well as outcome measures in an appropriate manner.

Cannon-Bowers and Salas (1997) argue that TPMs must consider measurement at the individual and team levels because both teamwork and taskwork skills influence team performance. Additionally, TPMs must include measures that address process as well as outcome. The process measures describe the activities, strategies, responses and behaviors relevant to the human that are used to accomplish a task. In the past, researchers have used several instruments to assess and measure process and outcome measures for operator actions at both individual and team level. Smith-Jentsch and her colleagues (Smith-Jentsch, Johnston and Payne 1998) provide a list of such instruments including: sequenced actions and latencies index (SALI), behavioral observational booklet (BOB), anti-air teamwork performance index (ATPI) and anti-air teamwork observation measure (ATOM). While SALI and BOB are measures used to evaluate individual level outcomes and processes, ATPI and ATOM are used to evaluate team level outcomes and performance. These instruments are used by experts in the field to provide subjective ratings for process and outcome measures at individual and team levels, and provide an indication of the expert's judgment of operator performance. Therefore, these ratings are subject to problems such as inter-rater reliability. Additionally, the subjective ratings provided by the experts are often decoupled from the objective measures of team performance.

In contrast to the existing measures listed above, we focus on a measure called the RAI (Thiruvengada and Rothrock 2007). RAI circumvents the inter-rater bias problem as it does not involve expert ratings. It is an instrument that provides an objective assessment of process and outcome measures based on time windows. Given the potential time window outcomes, RAI can be expressed as the ratio of the number of 'on time' correct actions executed by an operator for a class of time windows to the total number of time windows that are opened in that class for that specific operator role. The mathematical formulation for RAI is shown in Eq. 2.9.

$$\mathrm{RAI} = \frac{\text{Number of 'on time' correction actions for a class of time windows}}{\text{Total number of time windows that are opened in that class}} \tag{2.9}$$

In this study, team performance depends upon four teamwork dimensions: information exchange, communication, team initiative/leadership, and supporting behavior. The detailed explanation for each of these dimensions will be given later. Time windows that relate to a specific teamwork dimension, such as information exchange, are grouped together and are said to belong to the same class of time windows for calculating RAI.

2.2.2 Problem Domain

To demonstrate the utility of RAI, we conducted an empirical study with human participants using a human-in-the-loop simulation known as the *team Aegis simulation platform* (TASP). The objective of TASP is to reproduce a naval command-and-control environment in the combat information centre (CIC) task context (onboard a Navy ship with aircraft and missile launch capabilities) in which there are up to three operator roles functioning as a team, an anti air warfare coordinator (AAWC), an aircraft information coordinator (AIC) and a sensor operator (SO), acting concurrently. All operators have well defined tasks (responsibilities) set in a military context and are provided with rules of engagement (RoE) (Table 2.2) to help aid in their decision making process. The operators are recommended to follow the RoE at all times to achieve team goals. The RoE is different for each operator role in the team but governs their overall activities. Each operator is required to perform tasks based on RoE as well as compensate their teammates through supporting behavior (backup and error correction). The AAWC is the commander of team (team leader) and is responsible for coordinating the overall activities, including identifying unknown aircraft, assigning and engaging missiles on hostile aircrafts. The AIC is responsible for monitoring the activities of all friendly combat aircrafts, known as defensive counter air (DCA) and requesting visual identification (VID) report from them. The SO interprets any incoming sensor signals and issues warnings to hostile aircrafts violating the RoE.

There are several distinct as well as overlapping responsibilities among operator roles in TASP. At least one primary task responsibility on one role is shared among the other operator role, where the operator under whom the responsibility is listed has the primary action responsibility for that task. For example, the task of assigning primary identification label to any unknown aircraft is shared among the three roles, but the AAWC operator has the primary action responsibility for this task. Tasks on each role are executed through the use of a graphical user interface. As an example, Fig. 2.7 shows the graphical user interface for the AIC operator role.

The upper left box in Fig. 2.7 contains information about an object under consideration (e.g., an aircraft with an unknown identity). The spatial representation of objects in the vicinity of the AIC's ship is portrayed through the radar scope on the right half of the display. Action can be taken through the interface via function keys or buttons shown on the middle box in the left side of the display.

Table 2.2 Rules of engagement (RoE)

AIC	SO
1. Engage a Hostile aircraft within 20 nautical miles (NM) from ownship (hostile aircraft only). (AAWC RESPONSIBILITY BACKUP)	1. Issue level 3 warning to hostile aircraft only when it is within 20–30 nautical miles (NM). 2. Issue level 2 warning to hostile aircraft only when it is within 30–40 NM.
2. Assign a missile to a hostile aircraft within 30 NM from ownship (hostile aircraft only). (AAWC RESPONSIBILITY BACKUP)	3. Issue level 1 warning to hostile aircraft only when it is within 40–50 NM.
3. Maintain safety of DCA (e.g., keep DCA away from danger zones of hostile aircraft, do not let DCA run out of fuel, etc.).	4. Make a primary identification of air contact (i.e., friendly, hostile).[a] (AAWC RESPONSIBILITY BACKUP)
4. Keep DCA within 256 NM from ownship. 5. Keep DCA at least 20 NM away from ownship.	5. Evaluate, correlate and transmit all sensor value emissions that appear on the EWS interface.
6. Make a primary identification of air contact (i.e., friendly, hostile).[a] (AAWC RESPONSIBILITY BACKUP)	

[a] Once an aircraft has come within 50 NM from ownship, it should be identified before it travels an excess of 10 NM. If an aircraft "pops up" within 50 NM it should be identified before it travels an excess of 10 NM

Two overarching rules

(1) Defend ownship and ships in battle group

(2) Do not engage friendly or civilian aircraft

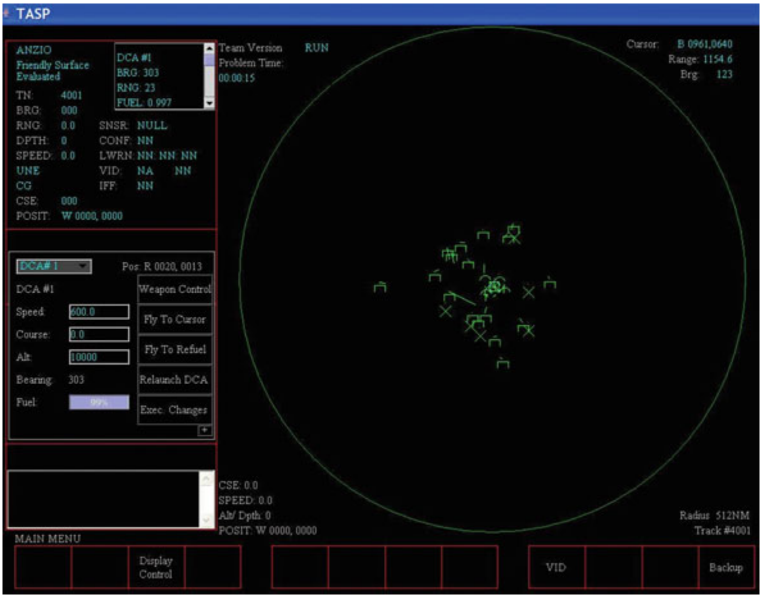


Fig. 2.7 Graphical user interface for an aircraft information coordinator (AIC) operator in TASP

For the purpose of this experiment, we consider a two person team of AIC and SO roles with no team leader AAWC. We use a simulation based approach that employs a truth maintenance system (TMS) in the background to keep tracks of information pertaining to time windows as well as operator actions. The time windows and operator actions data logged by TMS can be converted into a database using a converter tool in order to provide insights on metrics relating to RAI along teamwork dimensions. For example, when a hostile aircraft travels within 20 NM from the ownship, a time window is open specifying the opportunity of engaging the aircraft exists. When this aircraft travels out of the 20 NM range from ownship, the time window closes. If the AIC operator successfully engaged the aircraft within the time window, then AIC executed an ontime correct action. All these data are logged by TMS and the information can easily be queried from the database. The data can further be analyzed statistically to reveal the impact of the training intervention on team performance measures.

2.2.3 Teams and Performance Assessment Measures

Smith-Jentsch et al. (1998) defined four dimensions of teamwork for team dimensional training that are critical to overall team performance as information exchange, supporting behavior, communication and team leadership/initiative. Typically, these dimensions are assessed using post hoc surveys, questionnaires, and expert ratings. These dimensions are used to classify team outcome measures (time windows) into team process measures. While verbal communications existed between team members, specific content that was communicated between team members was not broken down to classify time window outcomes. Instead, time windows were opened and closed for each operator role based on the environmental conditions. These time windows are summarized in Table 2.3 and are classified into the teamwork dimensions.

The information exchange dimension relates to the process of gathering information and effectively exchanging them to develop a shared mental model for the team. Therefore, the AIC must fly a DCA within a certain distance to an unknown aircraft that approaches the vicinity of the ownship to gather its visual identification information (either friendly or hostile). In a similar fashion, the SO must detect sensor signal emissions and evaluate the intent of the signal as either friendly or hostile. The information that is gathered by both operators must be effectively exchanged among all team members. Therefore, time windows that are opened for visual identification and sensor evaluation process belong to this teamwork dimension. The tasks involved in this process are primary to the corresponding operator roles. Communication is external to the scope of this research as both operators exhibit implicit coordination without any overt communication. Both the AIC and SO operators must exhibit team initiative and leadership for the team's survival. Time windows pertaining to flying DCA out of potential threats and issuing warnings to approaching hostile aircrafts are classified under team

Table 2.3 Teamwork dimension classification of operator responsibilities for AIC and SO roles

Task type	Teamwork dimension	Responsibilities for operator roles	
		Aircraft information coordinator (AIC)	Sensor operator (SO)
Primary	Information exchange	Request visual identification (VID) report and pass it to other teammates	Evaluate incoming sensor signals
			Correlate sensor signal to a particular aircraft
			Transmit the correlated sensor signal
Backup	Communication	Operators did not use speech channels for communication (not considered)	Operators did not use speech channels for communication (not considered)
Primary	Team initiative/leadership	Vector defensive counter air (DCA) within 256 NM from ownship	Issue level one warning to hostile aircrafts
			Issue level two warning to hostile aircrafts
		Vector DCA outside 20 NM from ownship	Issue level three warning to hostile aircrafts
		Vector DCA outside danger zones. (Vectoring of DCA is done by changing its speed, course and altitude)	
Backup	Supporting behavior	Assign identification to unknown aircrafts	Assign identification to unknown aircrafts
		Assign missiles to hostile aircrafts	
		Engage missiles upon hostile aircrafts.	
Error correct-ion		Change the identification of incorrectly identified aircrafts	Change the identification of incorrectly identified aircrafts

initiative/leadership dimension for AIC and SO, respectively. Finally, identifying the unknown aircraft and error correcting incorrect identifications are part of the supporting behavior dimension for AIC and SO roles. AIC is also responsible for supporting AAWC role by assigning and engaging a missile on hostile aircrafts that pose a high threat within close proximity to the ownship.

Consider the following example to translate behavior data into RAI outcomes. Suppose performance data is collected from a scenario as shown Table 2.4. Based on the classification of operator responsibilities along with teamwork dimensions (see Table 2.3), RAI scores can be calculated for each dimension. For example, AIC's performance on the SB dimension involves two types of task activities, which are primary identifications (Primary ID) and assign and engage (A&E). We then can calculate RAI (AIC on SB) = [RAI (AIC on Primary ID) + RAI (AIC on A&E)]/

Table 2.4 Performance data from a sampled scenario

AIC								
Primary identification		VID		Assign and engage		RAI		
Opened time windows	On time correct actions	Opened time windows	On time correct actions	Opened time windows	On time correct actions	IE	TI/L	SB
16	2	2	2	1	0	01	NA	0.0625
SO								
Primary identification		Sensor operation		Issue level warnings		RAI		
Opened time windows	On time correct actions	Opened time windows	On Time correct actions	Opened time windows	On time correct	IE	TI/L	SB

2 = (2/16 + 0/1)/2 = 0.0625. In the same manner, the performance data for each group of participants can be collapsed to get their respective set of RAI scores.

2.2.4 Methodology

Participants in this research were first-year graduate and junior and senior-level undergraduate students at the Pennsylvania State University. A total of 78 students (39 two person teams), between the ages of 18 and 25, participated in this study. Of the total, 46 were male and 32 were female. They were skilled computer users and did not have any disabilities that restricted them from adequate use of mouse/ keyboard interface. Additionally, the participants did not have any prior experience with the simulation environment. The participants engaged in a single session that lasted for about 3.5 h on average, and were provided with monetary compensation at the end of the study.

The two independent variables used in the study include training and workload. No training (NT), team coordination training (TCT) and task delegation training (TDT) are used as the training conditions. In NT condition, team members are not trained with any teamwork skills. They are required to read an article that illustrates the utility of team coordination and task delegation. However they are not provided with any information that prescribes how team coordination and task delegation can be achieved. Team members in the TCT condition are provided with excerpts of coordination strategies, which includes monitoring designated areas and passing information to other teammates as needed. The training helps team members in creating a shared mental model and allows them to anticipate the expectancies of their teammates. In TDT condition, specific tasks are delegated to the team members. The AIC is delegated tasks relating to assigning and engaging upon hostile aircrafts with missiles and issuing identifications based on visual identification information. The SO is delegated with tasks relating to assigning identifications based on sensors values that are evaluated. Differences between

Table 2.5 Types of training

No training (NT)	Team coordination training (TCT)	Task delegation training (TDT)
No specific training is imparted	Team coordination is emphasized during training	Task delegation is emphasized during training
Team members are provided with information on the definition of team coordination and task delegation	Team members are instructed on how to achieve effective coordination via demonstration of good and bad practices	The radar scope on the operator's GUI is split into two distinct areas and is designated to each of the two roles. Operators monitor and perform actions within the designated area, while passing information pertaining to the other area onto their team mate
No specific tasks are delegated to each operator role	No specific tasks are delegated to each operator role	Specific tasks are delegated to each operator role based on KSA competencies and operator capabilities

training interventions are listed in Table 2.5. Workload stress levels are controlled by setting them at low and high levels. Different scenarios were developed for setting the stress levels of workload. The high stress workload scenarios include a relatively high number of hostile aircrafts that must be identified, assigned and engaged with missiles for both members within the team than in low stress workload scenarios.

Thirteen teams (one-third of 39 total teams) randomly received one of the three training conditions. The team members were randomly assigned to AIC or SO role. Each team was subjected to scenarios with both low and high workload stress levels.

The participants underwent an initial training of specific skills, which lasted for about an hour. This initial training enabled them to acquire skills that are necessary to accomplish tasks that are specific to their current roles. Four practice sessions (practice sessions 1–4) of 10 min duration each were provided to the participants to hone their role specific skills. During these practice sessions, the participants were given feedback on their performance relating to taskwork skills and were encouraged to ask any clarification questions. At the end of the four practice sessions, the teams were subjected to the first learning evaluation session for a duration of 10 min, which assessed their learning on taskwork skills. During this session, each team member was assigned specific tasks that would require them to use their taskwork skills and feedback about their performance was provided at the end of the session. After taskwork skills training, the teams were randomly exposed to one of the three team training interventions. In NT intervention, there was no hands-on training provided to the team regarding teamwork. Instead, they were instructed to read articles that explained the importance of teamwork and coordination. In "team coordination training" or TCT intervention, the teams were presented with instances of good and poor team coordination policies and were exposed to a video that demonstrated the same. In TDT intervention, the teams were provided with a presentation of different tasks that were delegated to their roles as part of the training intervention and were also shown a video that demonstrated teamwork associated with task delegation. After the appropriate training intervention was provided, the teams were given an opportunity to practice teamwork skills through two 10 min practice sessions. Then, the teams were exposed to a second learning evaluation session that assessed their teamwork skills. The teams were instructed to perform tasks that required the effective use of taskwork and teamwork skills. The teams were then subjected to two sessions (of 10 min duration each) with low and high stress levels of workload where data relating to the performance of each team member (SO and AIC) were collected for further analysis.

2.2.4.1 The Statistical Model

The linear regression of team performance is modeled such that:

$$Y_i = \sum_{j=1}^{J} X_{ij}\beta_j + \varepsilon_i, \text{ where } \varepsilon_i \sim N(0, \sigma^2). \tag{2.10}$$

In such a model we assume that the error terms are normally distributed, zero mean and the same variances for all cases. However, outcomes that are proportions, as are the RAI's yield a distribution which violates the normality and homoscedasticity assumptions. Accordingly, analyzing proportions with linear regression may lead to misleading inference about the explanatory variables. This led researchers to consider logistic regression as the model for analyzing data in which the dependent variable is a proportion. The logistic regression is modeled as:

$$E(Y_i) = \mu_i = p_i = \frac{\exp(\sum_{j=1}^{J} X_{ij}\beta_j)}{1 + \exp(\sum_{j=1}^{J} X_{ij}\beta_j)}, \tag{2.11}$$

where, $E(Y_i) = p_i$.

Equation 2.11, that can also be expressed as:

$$\log \frac{p_i}{1 - p_i} = \sum_{j=1}^{J} X_{ij}\beta_j \tag{2.12}$$

is a particular case of the *Generalized Linear model*, in which linear regression models are extended to the exponential family of distributions that includes both the normal and the binomial distributions. Such models involve a link function which is some transformation g(.) that linearizes the expected value of Y_i, such that $g(\mu_i) = \eta_i$, and $\eta_i = \sum_{j=1}^{J} \beta_j X_{ij}$ is a linear combination of the predictors. The normal error regression model is a generalized linear model with the identity function as the link function, such that $\mu_i = \eta_i$. For logistic regression model

Table 2.6 Raw means and predicted means of the experimental data

Teamwork dimension	Training intervention						Workload stress			
	NT		TCT		TDT		Low stress		High stress	
	R[a]	P[b]	R	P	R	P	R	P	R	P
IE	0.2550	0.4556	0.4008	0.563 1	0 .3178	0.5039	0.3385	0.5545	0.2694	0.4607
TI/L	0.1714	0.1465	0.2434	0.2280	0.2823	0.2586	0.2697	0.2425	0.195	0.1795
SB	0.1549	0.1346	0.1629	0.1327	0.1740	0.1407	0.1704	0.1426	0.1574	0.1296

[a] R is the raw mean from the observed data,
[b] P is the predicted means by the model

Table 2.7 Type III test of fixed effects

Teamwork dimensions	Training intervention		Workload stress		Interaction	
	df	F	df	F	df	F
IE	(2, 36)	3.76[a]	(1, 36)	13.94[b]	–	–
TI/L	(2, 35.68)	1.05	(1, 72)	2.66	(2, 72)	4.48[a]
SB	(2, 36)	0.81	(1, 36)	4.04[a]	–	–

[a] $p < 0.05$, [b] $p < 0.01$

$g(p) = \log \frac{p}{1-p}$, which is known as the logit function. Our experiment was designed to evaluate the effect of a certain type of training on an outcome Y, which is the proportion RAI. Since the dependent variable (RAI) is a proportion, the suitable distribution for modeling it, is the binomial distribution. The dependent variable Y in our experiment, was measured for each one of the two team members, at two stress levels (low/high), where each team belonged to one of three training groups (NT, TCT, TDT). The main aim in analyzing the data is to compare the groups on the outcome (RAI). For each of the 39 teams, divided randomly among the three types of training, there are four dependent measures of RAI since each team member (SO and AIC) has two outcome measures, corresponding to high and low levels of stress.

In the inference based on linear as well as generalized linear models, it is assumed that the observations are independent. Extending these models to account for correlated data led to the development of mixed models, for normal data, and more generally, to Generalized linear mixed models for the generalized linear models. Details of the model can be found in Rothrock et al. (2009).

2.2.4.2 Analysis and Results

The raw mean values and predicted mean values are shown in Table 2.6. Statistics of type III test of fixed effects are summarized in Table 2.7. The detailed analysis and results are elaborated for each teamwork dimension.

Table 2.8 Estimated covariance matrix for training and information exchange behavior

	AIC low	AIC high	SO low	SO high
AIC low	1.222 (0.285)	0.438 (0.233)	0.443 (0.283)	0.154 (0.403)
AIC high		1.286 (0.296)	0.311 (0.264)	−0.280 (0.428)
SO low			1.522 (0.382)	0.534 (0.356)
SO high				1.771 (0.590)

Standard errors are in parentheses

In the following, the standard errors (SE) of each estimate are displayed in brackets.

Training and Information Exchange

The analysis was performed based on 130 observations (26 were dropped due to zero value in the denominator). Since the interaction training × stress was found to be insignificant, it was dropped out from the model. The estimated covariance matrix for the experiment is shown in Table 2.8. From this matrix, we can observe the relationships of team member's performance (AIC and SO) on different stress workload levels (low and high). Though not significant, we observe a negative correlation between the AIC and SO in the high stress condition. We also observe higher variances for SO, compared with the AIC.

The results indicate significant differences between the two training conditions TCT and NT ($p = 0.01$). The estimated RAI for TCT and NT were 0.563 (SE = 0.029) and 0.456 (SE = 0.028), respectively. Additionally, significant difference were found between the two stress levels ($p = 0.0007$), where the estimated RAI was 0.554 (SE = 0.017), for the low level of stress and 0.461 (SE = 0.025), for the high level.

Training and Supporting Behavior

The analysis was performed on 156 observations (no missing values).

Here too, the interaction training × stress was found to be insignificant, therefore it was dropped out from the model. The estimated covariance matrix for the experiment (Table 2.9) indicates negative and significant correlations between the AIC and SO both in the high and low stress conditions. In other words, when the RAI of the AIC was higher than average, the corresponding RAI of the SO was lower than average. A positive and significant correlation is observed between the low and high stress for each member. In other words, when a team member was higher/lower than average in one stress condition he was also higher/lower than average in the other stress condition. The results also indicate higher variances in the low stress condition, where the low stress variance of the SO was even higher than that of the AIC.

Table 2.9 Estimated covariance matrix for training and supporting behavior

	AIC low	AIC high	SO low	SO high
AIC low	3.058 (1.055)	2.012 (0.781)	−4.668 (1.205)	−1.449 (0.583)
AIC high		1.949 (0.702)	−2.835 (1.082)	−1.073 (0.358)
SO low			9.880 (2.683)	2.774 (1.254)
SO high				1.955 (0.666)

Standard errors are in parentheses

No significant difference was found among the training levels ($p = 0.81$), yet a significant difference was found between the two stress levels. The estimated RAI least-squares mean (lsmean) was 0.143 (SE = 0.006) for the low stress, while it was only 0.130 (SE = 0.007), for the high stress ($p = 0.05$).

Training and Team Initiative/Leadership

The time windows data indicated none of the AIC operators were able to perform the DCA manipulations in the experiment. Therefore we only have data corresponding to the SO (78 observations). Nevertheless, in order to allow a correlation between the two conditions measured for the same person, a repeated measures structure was used. The intra-class correlation, indicating the correlation within each team member (i.e., the correlation between two observations that belong to the same team member) was high (0.874).

For this outcome variable, the interaction between stress and training was significant, ($p = 0.013$). There are six different combinations of stress with training which led to 15 pairwise comparisons. Among these 15 tests, three were found to be significant. The most significant was the difference between the stress levels in the TCT training condition. The estimated RAI lsmean was 0.346 (SE = 0.093) for the low stress, and only 0.11 (SE = 0.039) for the high stress ($p = 0.002$). A significant difference was also found between the two training conditions TCT and TDT in the high stress condition ($p = 0.03$). While the estimated RAI lsmean was only 0.11 (SE = 0.039) for the TCT it was 0.288 (SE = 0.070) for TDT. Finally, a significant difference was also found between the NT group in the high stress and the TCT group in the low stress ($p = 0.04$), where the estimated RAI lsmean was 0.346 (SE = 0.093) for the low stress TCT and only 0.14 (SE = 0.046) for the high stress NT.

2.2.5 Discussion

The statistical analysis revealed an interesting view of team performance. Under the information exchange dimension—where information about the visual identity and sensor signature of tracks is shared—we found that TCT training significantly improved performance. Moreover, we also noticed a trend toward a negative correlation between the AIC and the SO under stress, which suggests that teams tend to depend on a single source of information (either visual identification from the AIC or sensor information from the SO).

For the supporting behavior dimension, the effects of stress are more pronounced. A closer look at the type of activities involved with supporting behavior showed that they required longer key sequences to execute and that, under stress, fewer identification assignments were made. More importantly, as one role took on more activities under stress, the other role executed fewer activities. Therefore, just as information exchange tended toward uncertainty (i.e., only one source vs. two sources of information), supporting behavior also tended toward brittleness (i.e., one person assigning identities vs. two people).

In the team initiative and leadership dimension, our analysis discovered two interesting findings. The first is the absence of DCA activities, which suggests that the AIC either did not have the cognitive resources available to manipulate the DCA assets, or that the teams were not sufficiently trained to do so. In any case, the only data we had was the issuance of level warnings by the sensor operator. The second interesting finding was that participants exposed to TCT outperformed participants trained under either TDT or NT conditions. While the effect of the training was not universal across all stress combinations, our analysis suggests that TCT was more effective under high stress conditions. The comparison between the effects of TCT and TDT under the high stress condition was especially telling because TDT was developed to routinize responsibilities so that the effects of stress are mitigated.

2.3 Performance Assessment in an Interactive Call Center Simulation

In this part of the paper, a new performance assessment methodology call center systems at the level of customer-agent interactions (CAI) is proposed. A team-in-the-loop simulation test bed has been developed to analyze CAI-level performance using time windows. The proposed framework should allow researchers to collect and analyze individual as well as team performance at a finer granularity than current call center efforts.

2.3.1 Introduction

Today, we live in a service-based economy which faces challenges to assess and manage the performance of human-in-the-loop service systems (Chesbrough and Spohrer 2006). A case in point is the telephone call center which requires customer interactions for its operation. Because it is normally the first touch point of a business with which customers make contact, impressions on the total service

Table 2.10 Queue-centered call center measures

Measures	Description
Average speed of answer	The average time taken: for the call to be picked up
Average talk time	The average time that acaller waited to be connected to an agent
Queue time	The amount of time taken for a caller to wait in the line
Calls per hour	The average number of calls that an agent handles per hour
Hold time	The average time taken for an agent to place a customer on hold
Occupancy	The average time taken for an agent in his or her seat
Blocked calls	The total number of busy and out-of-order telephone trunks that block calls
Abandonment rate	The percentage of callers who disconnect prior to be answered
First call resolution	The percentage of calls closed on the first connect
Service level	Transactions that must be handled within given time frame

quality can be made from call center interactions. Traditionally, quality assessment has been made through direct call monitoring for every agent, which consumes tremendous amount of resources and times. In this regard, the proper modeling and evaluation of service systems can enable managers to effectively monitor service performance (Fleming et al. 2005).

Generally, a call center consists of trained customer service agents who answer customers' calls and coordinate their requests. Call center systems can provide a variety of functions such as help desk support, customer service, technical support, contact centers service, and tele-marketing etc. In this paper, we specifically focus on inbound call centers in which agents' assistance is sought by callers. Inbound call centers are very labor-intensive systems with high agent turnover rates amounting to "typically comprising 60–80% of the overall operating budget" (Aksin et al. 2007; Gilmore and Moreland 2000; Wallace et al. 2000). For this reason, managers tend to make an effort to improve the effectiveness of interaction between agents and customers through proper training and performance evaluation. Therefore, it should be priceless that managers can get a framework to provide quality information on their agents and customers interactions.

In previous research, the performance analysis of call centers has been mostly performed by using Erlang formulas that were designed for traditional queueing systems (Mehrotra and Fama 2003; Gilmore and Moreland 2000; Tanir and Booth 1999). These queueing based models may be useful and provide plentiful gross-level metrics in the case of evaluating the service performance in quantity assessments, as most call center research (Gans et al. 2003; Garnet et al. 2002) consider call centers as queueing system which consists of customers (callers), servers (telephone agents), and queues. Using this queue-centered approach, a variety of measures can be acquired and a representative sample of key performance indicators from Anton (1997) is shown on Table 2.10. Above all, the measure of a *telephone service factor* or *grade of service*, which is the percentage of calls answered in a given time frame, is widely used as a core measure (Sharp 2003). The previous works presented above, however, are only focused on quantity measures at a gross-level, while neglecting metrics of customer-agent interactions which specify the service quality within an individual service activity of call center operations. Aksin et al. (2007) also noted that a macro research theme such as "improving the way in which the tension between efficiency and quality of service is modeled" is significant for future call center operations research. Therefore, one can no longer simply equate service quality with customer waiting times.

While queue-centered analytic models are still popular, Mehrotra and Fama (2003) noted that several factors such as complex call traffic, rapid change operations, and cheaper and faster computing, have recently increased the demand for analysis of ever more complex call centers through simulation. Although there are simulation approaches which deal with call center problems based on the optimization such as linear programming and scheduling (Avramidis et al. 2009; Atlason et al. 2004; Cezik and L'Ecuyer 2008), still they focus on gross-level metrics. However, in order to provide training feedback and manage call centers effectively with proper performance metrics, managers should know the quality of interactions between agents and customers during services.

To address the limitations of exiting analytic queue-centered approaches, this paper presents a configurable help desk call center team-in-the-loop (TITL) simulation test bed called the call center workforce simulation platform (CCWSP), which is the interactive simulation framework for performance analysis at the team as well as individual-task levels. The proposed framework uses time windows to develop a performance measure at the CAI level. Specifically, a new metric is proposed, called the index of interactive service performance (IISP), to measure service quality at CAI level with consideration of temporal service success rates within service operations. CAIs are expressed as pre-defined time windows and can be mapped to gross-level measures.

2.3.2 Human-in-the-Loop Discrete-Event Simulation

A human-in-the-loop simulation provides both realistic as well as controllable interactive task environments. With a human-in-the-loop simulation, users interact in real-time with the simulation through a graphical interface, and we can directly gather user data in a controlled experimental environment. In many service applications, however, agents in the systems may work as a team as well as individually. The team performance can be much more important than an individual performance when designing service operations with human inclusions.

2.3.3 The Proposed Framework: Call Center Workforce Simulation Platform with Time Windows

In this section, a framework for the CCWSP based on an interactive TITL simulation is presented. The Information Technology Services (ITS) help desk at Penn State is modeled as a problem domain of the simulation. The CCWSP software

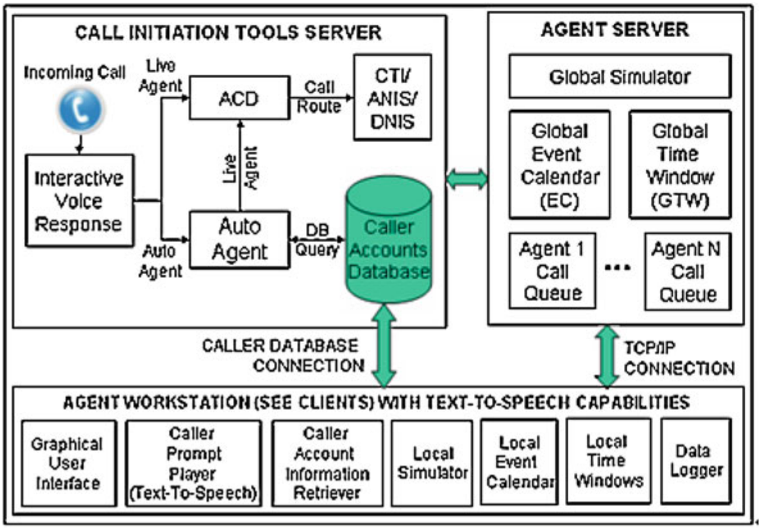


Fig. 2.8 Framework of the call center simulation application

architecture, the roles of time windows, and the call center simulation module are explained. In this discussion, we refer to the customer who makes a call to the call center as the caller and the service provider who answers a call as the agent.

The inbound help desk at Penn State University is used as the problem domain. The university ITS office runs the help desks system to handle calls by the Penn State user community on technical problems related to their computer hardware, software, network, and user account. In order to understand the help desk process, the operations of ITS help desk call center were analyzed through detailed field observations and a task analysis.

The application architecture for the help desk call center is built upon an interactive TITL discrete-event simulation that is comprised of three major parts: *Call Initiation Tools Server*, *Agent Server,* and *Agent Workstation* as shown in Fig. 2.8. The *Call initiation server* is a software component that provides an interface for live calls through computer telephony integration (CTI) equipment and updates the simulation about information pertaining to incoming calls. Currently, the server is driven by a pre-defined script file that simulates the caller information based on predefined scenarios. The *Agent Server* plays the role of a central server, not only in synchronizing the updates between various agents but also in placing a call in the caller queue as well as tracks gross-level performance metrics which can also be obtained from traditional queue-centered approaches. This server also maintains and tracks all individual workstation events through a global event calendar as well as windows of opportunity that exist for taking an action. The *Agent Workstation* simulates the events (based on a local event list and

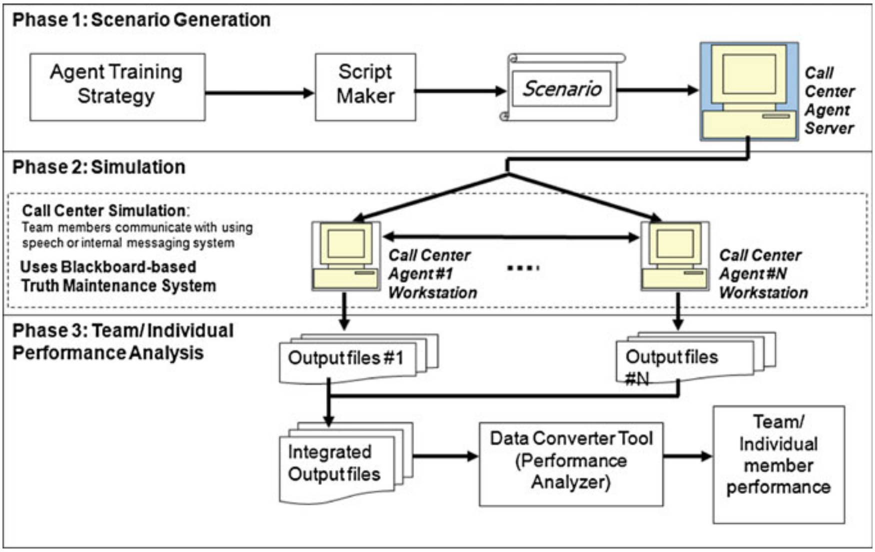


Fig. 2.9 Task flow diagram for the call center simulation framework

time windows) that are rendered on the graphical user interface (GUI) on each agent's workstation.

Three distinct phases collectively contribute to the performance measurement within the simulation framework. Figure 2.9 shows the task flow diagram of the three phases of the call center simulation framework. In Phase 1, the script files required to run the simulation are generated and initialized within the call center agent server. In Phase 2, the scenarios generated during Phase 1 are executed in real-time networked mode on each call center agent's workstation. This allows researchers to capture and log their actions into various data logs for further analysis. Finally, in Phase 3, the raw simulation output files are converted into a relational database for further performance evaluation.

In call center environments, the operations of time windows are not as restrictive and critical as those in command-and-control environments. Instead, a single operation is simply considered a link in the chain of the agent's activities required to perform a service call. For instance, if two consecutive actions (e.g., authentication and update a record) are needed to finish one service call, the situation of updating problems would be triggered by the agent's authentication action. On the other hand, in command-and-control operations, external factors based on rules of engagement, such as distance, altitude, and speed in a military radar system can situate agents' actions. Therefore, instead of specifying time duration for each time window, the opening and closing states are defined by agent's actions except in the case of a call drop. The latency in the agent's action is measured by the duration of each time window. As a result, only on time-correct actions, on time-incorrect actions, false alarms, and missed actions in Fig. 2.4 are

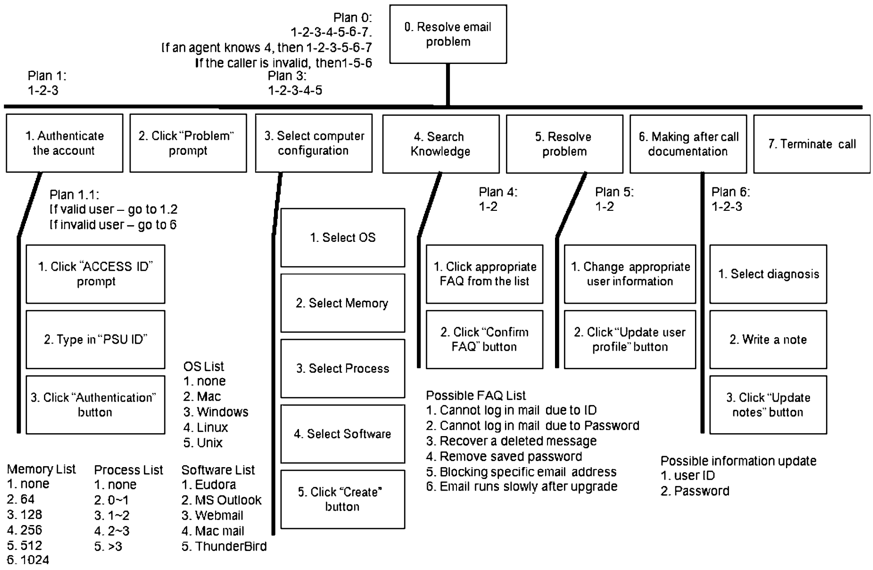


Fig. 2.10 HTA for a "Resolve email problem" task

possible results in the help desk simulation. The information of time window is logged for further performance analyses.

In call center environments, one call might be serviced by more than one agent (e.g., general agents can transfer technical calls to responsible agents). To handle this situation in a proper way, two levels of time windows are managed as global and local time windows. The *Agent Server* deals with global time windows to trace transferring calls, and the *Agent Workstation* takes care of each agent's local time windows.

To demonstrate the multi-level time windows in call center environments, "Update PSU account" and "Resolve email problem," are used as required tasks in our example domains. The result of hierarchical task analysis (HTA) of "resolve email problem" is illustrated in Fig. 2.10. This task is required to solve problems related to emails and has similar steps with the "updating PSU account" task except the "Selecting computer configuration." Both tasks need to be authenticated by caller's PSU ID. Once the caller's PSU ID is valid, the agent can proceed to communicate to figure out the problem by clicking the "Problem" prompt. Or the agent can go to the step of making an after-call documentation directly and indicate the caller is invalid. Next, the use of knowledge base will be determined based on the level of agent's expertness to provide proper information. By clicking the "Confirm FAQ" button, the agent determines whether the agent searches the right information. After changing the user's profile and clicking the "Update user profile" button, an after-call documentation with a proper diagnosis should be made by clicking the "Update notes" button. Finally, the agent can terminate the call. Within the two hierarchical tasks, 12 specific actions such as updating a user ID or removing a saved password are available.

Based on this task domain, the list of time windows is formulated as shown in Table 2.11. We define two types of time windows, primary and error-correction time window. The primary time window indicates the first available opportunity for an action. If the agent wants to revise his or her previous actions, then the error correction time window will gather the information. For the tasks at hand, there are a total of 19 time windows. Table 2.11 provides a breakdown of 11 primary and eight error-correction time windows with the action outcomes and open/close conditions.

Figure 2.11 illustrates the sequence of tasks along with the opening and closing conditions for each time window. At first, a caller places an incoming call which opens an overall time window for the call. Next, agents can see this incoming call on their call stack. Then, one of the agents picks up the call, which opens an authentication time window until the agent authenticates the caller. Once the caller is authenticated, the agent is able to complete other tasks such as creating problem profile, searching solutions, verifying user ID, verifying password, and resetting the password information. The TMS (truth maintenance system) opens primary time windows for these processes until the related action is performed. For example, if the agent creates a problem profile, the primary problem time window is closed by TMS and a secondary error correction time window is opened and remains open until the call ends. When the error correction time window is opened, the agent can correct any previous incorrect actions and all such agent actions along with the time window information are recorded in the output files for further performance evaluation.

From the time window's structure, we can categorize agent performances. If a time window is opened but no related agent action exists, then such an action is treated as a *missed action*, as shown in Fig. 2.9. On the other hand, if a time window is not opened but an agent action exists, then such actions would be related to a *false alarm action*. Only a related action is taken when a time window is opened can the action be considered on time and correct.

After gathering the time window information, the agent's performance at the CAI level is evaluated during the data analysis phase. In comparison to the queue-centered measure which gives overall values of system performances, the time windows-based measure in call center systems would give more detailed performance information related with human-interactions and deeper insights for call center managers. Nonetheless, the *Agent Server* in this simulation framework provides queue-centered measures, too.

To analyze and evaluate the quality of services in the CAI level, the appropriate method to describe quantitative service-related parameters is necessary. In the existing time windows approaches, Rothrock (2001) provide two method of evaluating operator performance based on time windows. The first method of factor analysis represents a technique that reduces factors to evaluate which situations and operator actions can be aggregated into higher order factors. The second method, signal detection theory (SDT), is designed to reveal an individual's decision criterion and the sensitivity of an individual's detection performance. Based on these

Table 2.11 List of 19 time windows in the task domain

Name of time window	Type	Open condition	Close condition
Overall call	*Primary*	Calls comes in the simulation	Call drop or end
Authentication	*Primary*	Pick-up button clicked	Authentication button clicked or call drop
Computer profile	*Primary*	Authentication button clicked	Create profile button clicked or call drop
Error correction for computer profile	*Error correction*	Create profile button clicked	Call drop or end
Diagnosis for the problem	*Primary*	Authentication button clicked	Confirm button in knowledge base clicked or call drop
Error correction for diagnosis	*Error correction*	Confirm button in knowledge base clicked	Call drop or end
Change password	*Primary*	Authentication button clicked under proper problem ID	Updated in password text field and reset button clicked
Reset ID	*Primary*	Authentication button clicked under proper problem ID	Updated in user ID and reset button clicked
Change name	*Primary*	Authentication button clicked under proper problem ID	Updated in name test field and reset button clicked
Change phone number	*Primary*	Authentication button clicked under proper problem ID	Updated in phone number text field and reset button clicked
Change E-mail address	*Primary*	Authentication button clicked under proper problem ID	Updated in e-mail address text field and reset button clicked
Change address	*Primary*	Authentication button clicked under proper problem ID	Updated in address text field and reset button clicked
Error correction for Change password	*Error correction*	Updated in password text field and reset button clicked	Call drop or end
Error correction for reset ID	*Error correction*	Updated in User ID and reset button clicked	Call drop or end
Error correction for change name	*Error correction*	Updated in name text field and reset button clicked	Call drop or end
Error correction for change phone number	*Error correction*	Updated in phone number text field and reset button clicked	Call drop or end
Error correction or change E-mail address	*Error correction*	Updated in e-mail text field a and reset button clicked	Call drop or end
Error correction for change address	*Error correction*	Updated in address text field and reset button clicked	Call drop or end
Notes	*Primary*	Authentication button clicked	Update button clicked or call drop

two methods, Thiruvengada and Rothrock (2007) suggested the RAI to evaluate team performance in a command-and-control human-in-the-loop simulation. The RAI can give quick and quantitative measures of performance data for system evaluation.

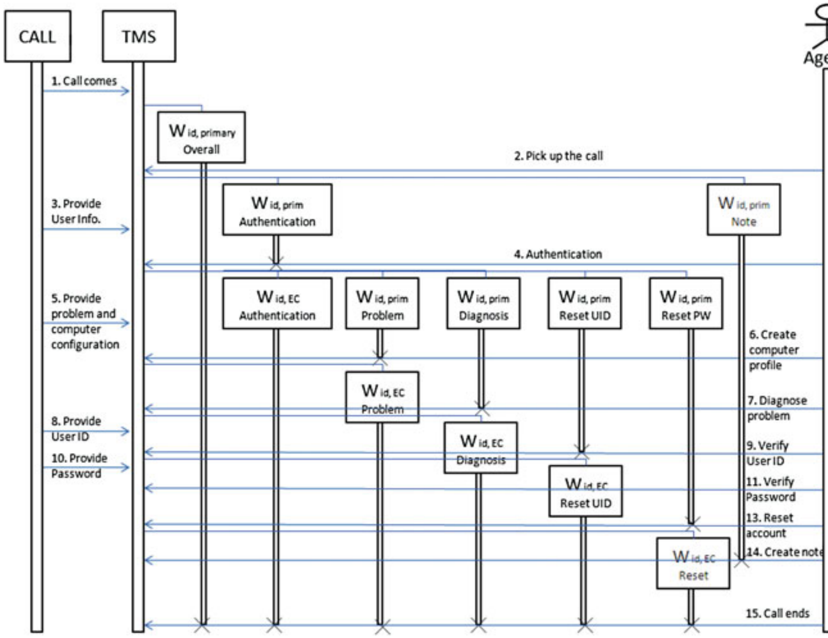


Fig. 2.11 Sequence diagram for time windows

However, the proposed index of RAI is difficult to be applied to measuring the CAI level metrics in service systems because of its strict adherence to time constraints in fixed-rule domain such as military operations.

To make the quantitative measures and analyze time windows-based measures in call centers by linking up with service quality and customer satisfaction, we suggest a new index, called an Index of IISP. We define IISP as an agent's ability to provide the correct service within a service level. The term service level refers to transactions that must be handled on arrival at the call centers. In this paper, the service level corresponds to the one of service quality standards and is expressed as the time limitation of the service. IISP is interpreted as the ratio of the number of "within service level" correct actions conducted by an agent for a class of time windows to the total number of time windows that should be opened in the class. Because there are two types of time windows, primary and error correction, in the list of time windows in Table 2.2, total time windows would be calculated as an average of the two types of time windows. The mathematical representation for IISP is formulated as follows, where SL_Correct is the area ① in Fig. 2.12:

$$\mathrm{IISP} = \frac{\sum_{i=1}^{n} \mathrm{SL\_Correct(i)}}{n} \tag{2.13}$$

(SL = service level, n = number of time windows)

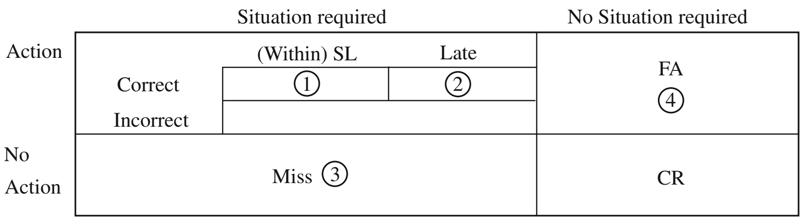


Fig. 2.12 Possible time window outcomes

Table 2.12 Metrics of human performance based on time windows

	IISP	RAI	Factor analysis	SDT
Purpose	Quantitative index for time windows in less strict domain (service domain)	Quantitative index for time windows in fixed-rule domain (command-and-control)	Determination of correlations among different time windows	Determination of the sensitivity of operator actions to situation requirements
Outcome	The ratio of the time windows number of "within service level" correct actions to the total number of time windows that should be opened	The ratio of the time windows number of ontime-correct actions to the total number of time windows that should be opened	Correlations among factors	ROC curve which is represent the receiver operating characteristic
Note	A guideline of IISP score is needed for subject matter experts	A guideline of RAI score is needed for subject matter experts	Screen test can be used	Accurate accounting method for correct rejections is needed

The comparison among the performance measures of time windows is shown in Table 2.12. The proposed IISP is for less strict domains such as service systems and can give users to judge the service quality and performance of the systems. For more details on a simulation study involving CCWSP, see Ma et al. (2011).

2.3.4 Discussion

The IISPs can show the degradation of service qualities while queue-centered gross level approach, which counts only on processing time, cannot capture the overall service performance in detailed levels. Time windows approach provides information of what kinds of specific sub-processes are required to be improved for either an individual agent or a group of agents.

The proposed framework consists of not only queue-centered measures but also CAI ones. In particular, small-sized call centers could benefit from IISP measures due to the large variance of agents' performance. For middle and large-sized call centers, the framework also provides benefits in terms of training and investigating agents' performance under interested situations. If the agents repeatedly miss or fail some time windows, then remediation can be the training of tasks that improve performance on those windows. Also, the framework enables managers to simulate specific situations or new service systems. For example, if a manager wants to know the effect of new call distribution system towards agents' performance, then he or she can compare the simulation results in the system.

In comparison with gross measures such as queue time and call duration (talk time), IISP is more diagnostic of individual tasks performed. Therefore, managers can easily understand both a system and workforce information with it. The detailed meaning of IISP would be captured from the raw time windows information. IISP also enables managers to compare their agents and help to generate a workforce performances profile.

In order to analyze time windows information from CCWSP, time windows must be categorized and defined clearly. For the ITS help desk task domain, 19 time windows were pre-defined. Also, managers need to set the service level correctly. Finally, by testing participants with the target scenario, time windows information can be gathered, and IISP can be calculated along with other queue-centered measures.

2.3.5 Conclusions

A research approach to evaluate operator performance in human-in-the-loop simulations has been proposed. The key concept within the approach is a notion of time windows. The time window construct provides a computational framework to dynamically evaluate operator actions in the context of heterogeneous task demands.

To implement time windows in a working model, a blackboard paradigm was introduced. The blackboard model is suited to accommodate the time window construct because of its ability to reason opportunistically about the availability of situations and the timeliness of operator actions. It was argued that human-in-the-loop simulations are ideal tools to investigate dynamic phenomena without concerns of the oversimplified laboratory environment or the unconstrained real-world. Therefore, requirements for implementation of the blackboard model were discussed. Moreover, a study which implemented the blackboard model in a human-in-the-loop simulation was used to illustrate the viability of the time window construct to provide a framework for operator performance. Two methods for analysis of time window outcomes were discussed to provide complementary perspectives on operator attunement to the constraints.

Time windows were then used to develop the RAI as a measure of team performance, a proposed standard that cuts across disciplines and enables the use of statistical techniques to aid researchers in better understanding team decision making. By using RAI as the primary metric, inter- and intra-rater reliability difficulties faced by the researchers are avoided. Ultimately, the effectiveness of an RAI-based measure is contingent on the ability of evaluators to establish the rules that govern a particular task domain. For example, temporal rules in a command-and-control domain are fairly straight-forward to extract whereas rules in a political debate are much more difficult to obtain. In general, RAI-based measures are more effective in domains where standard operating procedures and time constraints are clearly defined.

Finally, RAI was extended to a service enterprise—the call center. An approach using time windows-based assessment of an inbound call center system was proposed, which enables researchers not only to explain queue-centered measures utilized by most call center researchers, but also to explicate CAI measures. A configurable Team-in-the-loop simulation of a help desk, the CCWSP, was used to demonstrate the utility of this methodology. We also suggested a new quantitative index of agent performance, the Index of IISP which can provide a quantitative analysis of the agent service performance based on time windows. From the IISP, time windows-based measures from CCWSP can be systematically analyzed.

References

Adeli H, Yu G (1995) An integrated computing environment for solution of complex engineering problems using the object-oriented programming paradigm and a blackboard architecture. Comput Struct 54(2):255–265

Aksin O, Armony M, Mehrotra V (2007) The modern call-center: a multi-disciplinary perspective on operations management research. Prod Operat Manag 16(6):655–688

Allen JF (1983) Maintaining Knowledge about Temporal Intervals. Commun ACM 26(11): 832–843

Anton A (1997) Call center management: by the numbers. Purdue University Press, West Lafayette

Atlason J, Epelman MA, Henderson SG (2004) Call center staffing with simulation and cutting plane methods. Ann Oper Res 127:333–358

Avramidis A, Chan W, L'Ecuyer P (2009) Staffing multi-skill call centers via search methods and a performance approximation. IIE Trans 41(6):483–497

Beach LR, Lipshitz R (1993) Why classical decision theory is an inappropriate standard for evaluating and aiding most human decision making. In: Klein GA, Orasanu J, Calderwood R, Zsambok C (eds) Decision making in action: models and methods. Ablex Publishing Corp, Norwood, pp 21–35

Booch G, Rumbaugh J, Jacobson I (1999) The unified modeling language user guide. Addison-Wesley, Reading

Caldeira JD (1980) Parametric assumptions of some "nonparametric" measures of sensory efficiency. Hum Factors 22(1):119–120

Cannon-Bowers JA, Salas E (1997) A framework for developing team performance measures in training. In: Brannick M, Salas E, Prince C (eds) Team performance assessment and measurement: theory, methods, and applications. Erlbaum, Hillsdale, pp 45–62

Cattell RB (1966) The meaning and strategic use of factor analysis. In: Cattell RB (ed) Handbook of multivariate experimental psychology. Rand McNally, Chicago

Cezik MT, L'Ecuyer P (2008) Staffing Multiskill Call Centers via Linear Programming and Simulation. Manag Sci 54(2):310–323
Chesbrough H, Spohrer J (2006) A research manifesto for services science. Commun ACM 49(7):35–40
Chu RW, Jones PM, Mitchell CM (1991) The Georgia tech payload operations control center simulation: design and implementation in C++. In: Proceedings of the object-oriented simulation track of the 1991 society for computer simulation multiconference, Anaheim, CA
Coovert MD, Cannon-Bowers JA, Salas E (1990) Applying mathematical modeling technology to the study of team training and performance. Paper presented at the 12th annual interservice/industry training systems conference
Craig A (1979) Nonparametric measures of sensory efficiency for sustained monitoring tasks. Hum Factors 21(1):69–78
Endsley MR (2006) Situation awareness. In: Salvendy G (ed) Handbook of human factors and ergonomics, 3rd edn. John Wiley and Sons, Inc, New York, pp 528–542
Erman DL, Hayes-Roth F, Lesser VR, Reddy D (1980) The HEARSAY-II speech understanding system: integrating knowledge to resolve uncertainty. ACM Comput Surv 12:213–253
Fleming JH, Coffman C, Harter JK (2005) Manage your human sigma. Harv Bus Rev 83(7/8): 106–114
Gabbay DM, Hodkinson I, Reynolds M (1994) Temporal logic: mathematical foundations and computational aspects. Oxford University Press, Oxford
Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. Manuf Serv Oper Manag 5(2):79–142
Garnet O, Mandelbaum A, Reiman M (2002) Designing a call center with impatient customers. Manuf Serv Oper Manag 4(3):208–227
Gilmore A, Moreland L (2000) Call centres: how can service quality be managed? Ir Market Rev 13(1):3–11
Green PE (1978) Analyzing multivariate data. Dryden Press, Hinsdale
Green DM, Swets JA (1966) Signal detection theory and psychophysics. Wiley, New York
Greeno JG (1994) Gibson's affordances. Psychol Rev 101(2):336–342
Greeno JG (1998) The situativity of knowing, learning, and research. Am Psychol 53(1):5–26
Greeno JG, Moore JL (1993) Situativity and symbols: response to Vera and Simon. Cogn Sci 17:49–60
Hammond KR (1986) Generalization in operational contexts: what does it mean? can it be done? IEEE Trans Syst Man Cybern, SMC 16(3):428–433
Howie DE, Vicente KJ (1998) Measures of operator performance in complex, dynamic microworlds: advancing the state of the art. Ergonomics 41(4):485–500
Jones PM, Chu RW, Mitchell CM (1995) A methodology for human-machine systems research: knowledge engineering, modeling, and simulation. IEEE Trans Syst Man Cybern 25(7)
Kirlik A (1995) Requirements for psychological models to support design: toward ecological task analysis. In: Flach JM, Hancock P, Caird J, Vicente KJ (eds) Global perspectives on the ecology of human-machine systems. Erlbaum, Hillsdale, pp 68–120
Laudeman IV, Palmer EA (1995) The quantitative measurement of observed workload in the analysis of aircrew performance. Int J Aviat Psychol 5(2):187–197
Ma J, Kim N, Rothrock L (2011) Performance assessment in an interactive call center workforce simulation. Simul Model Pract Theory 19:227–238
Marks MA, Sabella MJ, Burke CS, Zaccaro SJ (2002) The impact of cross-training on team effectiveness. J Appl Psychol 87(1):3–13
Mehrotra V, Fama J (2003) Call center simulation modeling: methods, challenges, and opportunities. Paper presented at the 2003 winter simulation conference
Militello LG, Hutton RJB (1998) Applied cognitive task analysis (acta): a practitioner's toolkit for understanding task demands. Ergonomics 41(11):1618–1641
Moray N, Dessouky MI, Kijowski BA, Adapathya R (1991) Strategic behavior, workload, and performance in task scheduling. Hum Factors 33(6):607–629

Nii HP (1986) Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. Part I. AI Mag 7:38–53

Raby M, Wickens CD (1994) Strategic workload management and decision biases in aviation. Int J Aviat Psychol 4(3):211–240

Rothrock L (2001) Using time windows to evaluate operator performance. Int J Cogn Ergonomics 5(1):1–21

Rothrock L, Cohen A, Yin J, Thiruvengada H, Nahum-Shani I (2009) Analyses of team performance in a dynamic task environment. Appl Ergonomics 40(4):699–706

Rubin KS, Jones PM, Mitchell CM (1988) OFMspert: inference of operator intentions in supervisory control using a blackboard architecture. IEEE Trans Syst Man Cybern SMC 18(4):618–637

Sanderson PM, Verhage AG, Fuld RB (1989) State-space and verbal protocol methods for studying the human operator in process control. Ergonomics 32:1343–1372

Sharp D (2003) Call center operation: design. Digital Science Operation, and Maintenance, New York

Smith-Jentsch KA, Johnston JH, Payne SC (1998) Measuring team-related expertise in complex environments. In: Cannon-Bowers JA, Salas E (eds) Making decisions under stress: implications for individual and team training. APA, Washington, D.C., pp. 61–87

Smith-Jentsch KA, Zeisig RL, Acton B, McPherson JA (1998) Team dimensional training: a strategy for guided team self-correction. In: Cannon-Bowers JA, Salas E (eds) Making decisions under stress: implications for individual and team training. APA, Washington, D.C., pp 271–297

Suchman LA (1987) Plans and situated actions: the problem of human machine communication. Cambridge University Press, Cambridge

Swets JA (1996) Signal detection theory and ROC analysis in psychology and diagnostics: collected papers. Lawrence Erlbaum Assoc, Mahwah

Tanir O, Booth RJ (1999) Call center simulation in Bell Canada. Paper presented at the 1999 winter simulation conference. Phoenix, AZ

Thiruvengada H, Rothrock L (2007) Time window-based performance measures: a framework to measure team performance in dynamic environments. Cogn Technol Work 9(2):99–108

Vicente KJ, Rasmussen JR (1992) Ecological interface design: theoretical foundations. IEEE Trans Syst, Man Cybern 22(4):589–606

Vranes S, Lucin M, Stanojevic M, Stevanovic V, Subasic P (1991) Blackboard metaphor in tactical decision making. Eur J Oper Res 61:86–97

Wallace CM, Eagleson G, Waldersee R (2000) The sacrificial HR strategy in call centres. Int J Serv Ind Manag 11(2):174–184

Wickens CD, Hollands JG (2000) Engineering psychology and human performance, 3rd edn. Prentice Hall, Upper Saddle River

Wickens CD, Kessel C (1979) The effects of participatory mode and task workload on the detection of dynamic system failures. IEEE Trans Syst Man Cybern 9(1):24–34

Chapter 3
Virtual and Constructive Simulations with the GRBIL Modeling Tool

Michael Matessa and Walter Warwick

Abstract The Graph-Based Interface Language (GRBIL) tool combines aspects of virtual and constructive simulations. GRBIL can be used to set up a virtual simulation where people can interact with a simulation of an operator interface and environment. Human-in-the-loop activity can be recorded when a person performs a procedure with the simulated interface. This activity can then be automatically compiled into an operator model that can be used in constructive simulations where the operator model interacts with the simulated interface. The operator model can then make human performance predictions.

3.1 Introduction

Human-in-the-loop activity is normally associated with virtual simulations, simulations in which real people operate simulated systems. This activity can make use of human motor control skills or decision making skills. Virtual simulations can be considered in the broader spectrum of Live, Virtual, and Constructive simulations (DoD 1998). In this continuum, constructive simulations involve simulated people operating simulated systems. This stage can be useful in concept refinement and technology development. With the addition of real people, virtual

M. Matessa (✉) · W. Warwick
Alion Science and Technology, Boulder, USA
e-mail: mmatessa@alionscience.com

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_3,

simulations allow system development and demonstration. Further refinement of systems can lead to live simulations where systems are tested in operational settings. Human interaction is necessary in all three modes of simulation, with subject matter experts as sources of the knowledge engineering that goes into constructive simulations, as humans-in-the-loop in virtual simulations, and serving operational roles in live simulations.

This chapter describes how the Graph-Based Interface Language (GRBIL) tool (described in detail in Matessa and Mui 2009) combines aspects of virtual and constructive simulations. GRBIL can be used to set up a virtual simulation where people can interact with a simulation of an operator interface and environment. In this chapter, "interface" means the static physical layout of the system to be modeled, while the "interface simulation" includes the dynamic consequences of activating controls on the interface. Human-in-the-loop activity can be recorded when a person performs a procedure with the simulated interface. This activity can then be automatically compiled into an operator model that can be used in constructive simulations where the operator model interacts with the simulated interface. The operator model can then make predictions of operator motor control skills. In addition, a person can use GRBIL to specify the stimulus that triggers the execution of a procedure. This decision making skill is then captured by the operator model.

Although the GRBIL tool includes a human-in-the-loop during the development of an operator model, the ultimate intent is to take the human-out-of-the-loop during simulation. In this way, GRBIL supports the exploration of a much wider range of behaviors than would be possible in a virtual simulation; the execution of a constructive simulation is not limited by eagerness or availability of human subjects and it can be run under a variety of initial conditions to increase the likelihood of identifying low-frequency, high-consequence events. At the same time, by supporting the automatic generation of an operator model directly from human input, GRBIL leverages virtual simulation as a surrogate for knowledge engineering and thereby reduces the resources needed to develop and include principled models of human behavior in constructive simulations. This unique role of the human-in-the-loop in GRBIL is reflected in a system architecture that combines virtual and constructive simulation.

3.2 GRBIL Architecture

The architecture for GRBIL consists of four components, shown in Fig. 3.1:

1. The interface constructor—used to draw and define the GUI that the operator will interact with and to specify the operator's procedures.
2. The cognitive modeling system—responsible for generating predictions of human performance.
3. A dynamic environment model—responsible for modeling environments external to the interface that may be changing and whose changes will affect the operator's performance with the interface.

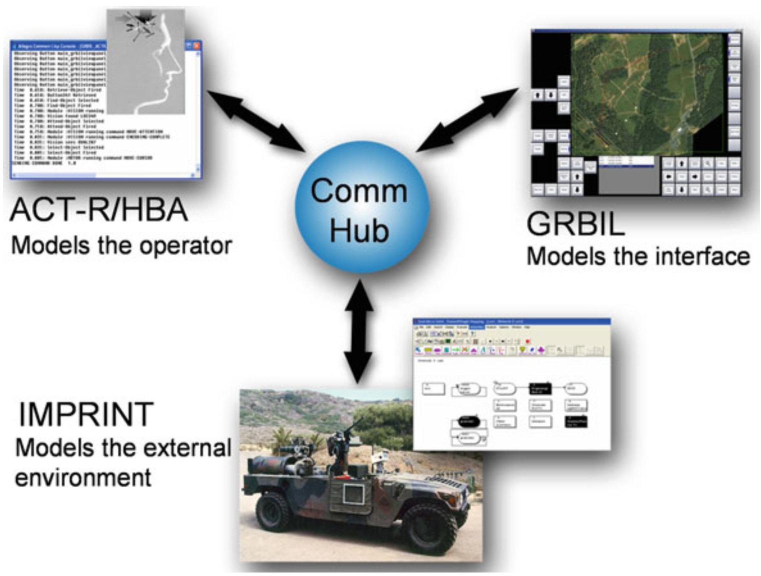


Fig. 3.1 The GRBIL system architecture

4. A software hub—used to mediate communication between the other components and to achieve the necessary level of component integration.

3.2.1 Component 1: Interface Constructor

The first component of GRBIL is the operator interface constructor, through which the analyst can describe the physical look and feel of the GUI being analyzed, as well as interface control actions. The interface constrains actions that humans-in-the-loop or, alternatively, a constructive operator model can take in simulations. Control placement is done in a similar fashion to many modern interface layout applications using WYSIWYG drag-and-drop functionality. An assortment of commonly used GUI widgets (e.g., radio button, text box, lists, and toggle buttons) can be dragged onto the GUI description. Once the control is placed on the GUI description, control properties can be used to adjust the labeling, size, shape, color, and behavior of the control.

The next step in describing a new GUI is to define the actions each control produces and what the desired effect of each action is. This is done for each control in GRBIL via an "Event Actions" menu for each control. Using this process of adding GUI windows, placing controls on those GUIs and then describing the effects of using those controls on the state of the GUI and all its windows, a GRBIL user can completely describe the functionality of a new user interface.

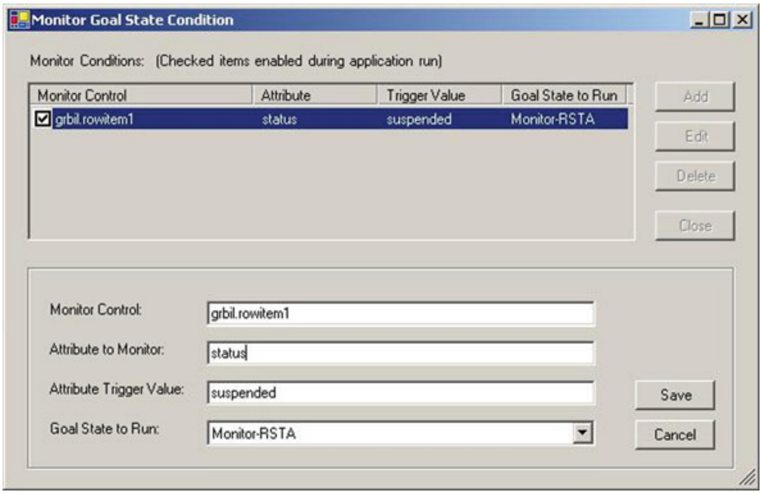


Fig. 3.2 GUI for the creation of monitors that trigger procedures

A byproduct of this work is that the GRBIL user has now developed a dynamic prototype of the new system's GUI. As a minimum, this prototype, once finalized, can be used as a "look and feel" specification for the project programming staff. In many cases, this prototype can actually be encapsulated and can function as the actual new system's software interface once connected to the appropriate action routines and system behaviors.

3.2.2 Component 2: Cognitive Modeling System

The second component of GRBIL is an embedded cognitive model that predicts how the human operator might behave when confronted with the GUI described in Component 1. The model is used in constructive operator simulations that require motor control and decision skills. Operator procedures are automatically compiled into the model by recording human-in-the-loop activity when a person performs a procedure with the interface in a virtual interface simulation. This is accomplished by stepping through and recording a series of actions, i.e., key presses, button clicks, mouse movement, etc. The recorded series of actions are associated with a "goal state." Many separate goal states can be defined for a given GUI, so that the GRBIL user can specify a robust set of potential user actions. Once specified, this goal state can function as the goal for the cognitive model embedded in GRBIL. (See Sect. 3.3.3 for a specific example).

Decision making skills are included by specifying stimuli to trigger procedures. As Fig. 3.2 shows, a GUI interface allows the selection of an attribute to monitor, the "trigger value" of the attribute, and the goal state to set when the monitor is triggered. The computational cognitive modeling architecture used in GRBIL is Adaptive Control of Thought-Rational (ACT-R) 6.0 which can be used in the original Lisp implementation (Anderson 2007) or in a task network implementation Human Behavior Architecture (HBA) (Warwick et al. 2008). ACT-R is a computational cognitive architecture that accepts declarative and procedural knowledge about how to do the task and, after combining it with a computational description of the environment in which the knowledge will be applied, can generate a time-ordered series of behaviors. These behaviors include cognitive tasks such as attention shifts, memory retrievals, and decisions. They also include motor effects, such as button presses or movements of the mouse. In this way, ACT-R can manipulate the interface as it attempts to achieve its goal state.

A number of characteristics of ACT-R are relevant to the performance of the model. First, ACT-R is limited in how fast it can perform its actions, especially external actions such as perceptual scanning and manipulation of the interface. In a dynamic, real-time environment such as robotic control, this can give rise to errors because the model is not able to keep pace with the demands of the task in the same way that a human operator would be unable to keep pace given the current system interface. Another source of errors is memory retrieval; ACT-R can skip steps or retrieve them in the wrong order (Anderson et al. 1998) in the same way that a human might. Finally, performance can vary as a function of individual differences in working memory, psychomotor speed, or individual strategies, all of which can be represented in a principled manner in the cognitive model. Because models are assumed to represent a steady state of practiced behavior, GRBIL does not take advantage of ACT-R learning mechanisms.

3.2.3 Component 3: A Dynamic Environment Model

The third component of GRBIL is the representation of the environment in which the new GUI will be used. This representation can be used in virtual interface simulations or constructive operator simulations. In many cases, system interfaces are not only responsive to inputs from an operator, but also reflect changes in the external environment, such as in an airplane cockpit. For this reason, we needed to design GRBIL to allow for the easy incorporation of systems which model the environment external to the GUI and the interface operator. Once incorporated into the GRBIL tool, this would enable changes in the environment model to be reflected on the GRBIL representation of the new GUI. This may affect the performance of the cognitive model, and vice versa. These requirements led us to select a task network modeling architecture for this component.

We selected the Improved Performance Research and Integration Tool (IMPRINT), developed by the ARL Human Research and Engineering Directorate (Archer and Adkins 1999), as the task network modeling environment for GRBIL. We chose this environment primarily because the discrete event simulation techniques included in IMPRINT are very well suited for human performance modeling. Secondly, IMPRINT is a stable software tool, originally developed to support the assessment of human performance in the context of total system performance in complex environments. IMPRINT provides a mature architecture and database structure that can easily incorporate a modeling method for representing goal-oriented behavior.

The basic modeling capability is a classical reductionist method. IMPRINT requires the decomposition of a system mission into functions which, in turn, are decomposed into tasks. The tasks are linked together into a network describing the flow of events. The network can include various types of branching logic such as parallel branches, probabilistic branches, and repeating branches. At the task level, estimates of task performance time and accuracy means and standard deviations are entered along with the consequences of the failure to perform a task accurately enough. The data entered are assumed to be representative of performance under "typical" or baseline conditions. Also, standards of performance can be entered to provide benchmarks for performance adequacy at the mission, function, and task levels. IMPRINT is very well suited to describe the events that could occur in the environment that will affect how the new GUI being analyzed in GRBIL must be used. For example, changes in terrain, the appearance of enemy or friendly units, or the availability of new information (e.g., mission orders, contact reports, intelligence data) could all change the way in which the GUI described in GRBIL would be used. While GRBIL does not contain IMPRINT, it does allow a user to link to a model developed in IMPRINT.

3.2.4 Component 4: Software Hub

This final component achieves the integration of and communication between the other three components at run time. Ensuring that multiple components produce a coherent simulation is a complex and difficult problem. In this case a software "hub" is used to arbitrate among the GRBIL components and advance the resulting human behavior simulation. As a GRBIL simulation runs and time progresses, the task network model provides event triggers that represent changes in the environment. These events could trigger changes in the GUI that the first component is showing (perhaps symbology on a map display changes, or perhaps a target is identified). In order to respond to these changes in the environment, the cognitive model must not only be executing a series of predefined procedures, but must also be monitoring the GUI for changes in the environment and making the appropriate response at the appropriate time.

At run time, GRBIL's second component (the ACT-R model) monitors the conditions at the end of each procedure. If the condition is found to exist, then the model takes the appropriate action. In this way, the ACT-R model is responsible for maintaining the ordered series of procedures and for managing and predicting how the simulated operator will process the available information.

The software hub enables the second and third components of GRBIL to run in parallel, so that changes to the environment can be happening at the same time that the modeled human is working on a current goal. This is necessary to provide realism, but also requires a fairly sophisticated time management capability within GRBIL. The software hub runs under the Windows operating system and uses socket connections for communication.

3.3 Simulation Modes

GRBIL can be used in four basic simulation modes: a virtual interface simulation, a virtual simulation with dynamic environment, a constructive operator simulation, and a constructive simulation with dynamic environment. Each mode will be described with an example application.

3.3.1 Virtual Interface Simulation

A fully specified interface simulation can be created in GRBIL that allows people to interact with interface controls and observe the control functionality. One example of such a simulation is an interface for an unmanned vehicle Operator Control Unit (OCU). The construction of this interface is seen in Fig. 3.3. The upper half of the figure shows how the OCU interface is laid out using control widgets from a palette. The lower half of the figure shows the GRBIL interface used to specify the effects of control actions taken with the OCU. In this case, the modeled functionality of the OCU includes the ability to switch display modes and to zoom the map view in and out.

Another example involves the design of a dismounted Tactical Control Unit interface for simplified hand-held control (Fig. 3.4). Using the GUI-based interface design capability of GRBIL, the spatial layout and functionality of the interface were quickly generated without writing any code.

3.3.2 Virtual Simulation with Dynamic Environment

With the addition of an IMPRINT model of the environment, the interface can show more dynamic information. As an example, the OCU interface is used to display unmanned vehicle location and, given an environment model, each vehicle can be moved independently (Fig. 3.5).

3.3.3 Constructive Operator Simulation

As described above, an operator model representing motor control skills can be easily created by demonstrating procedures using the virtual interface simulation. For example, to set up routes for UAV using the OCU, the procedure would be to

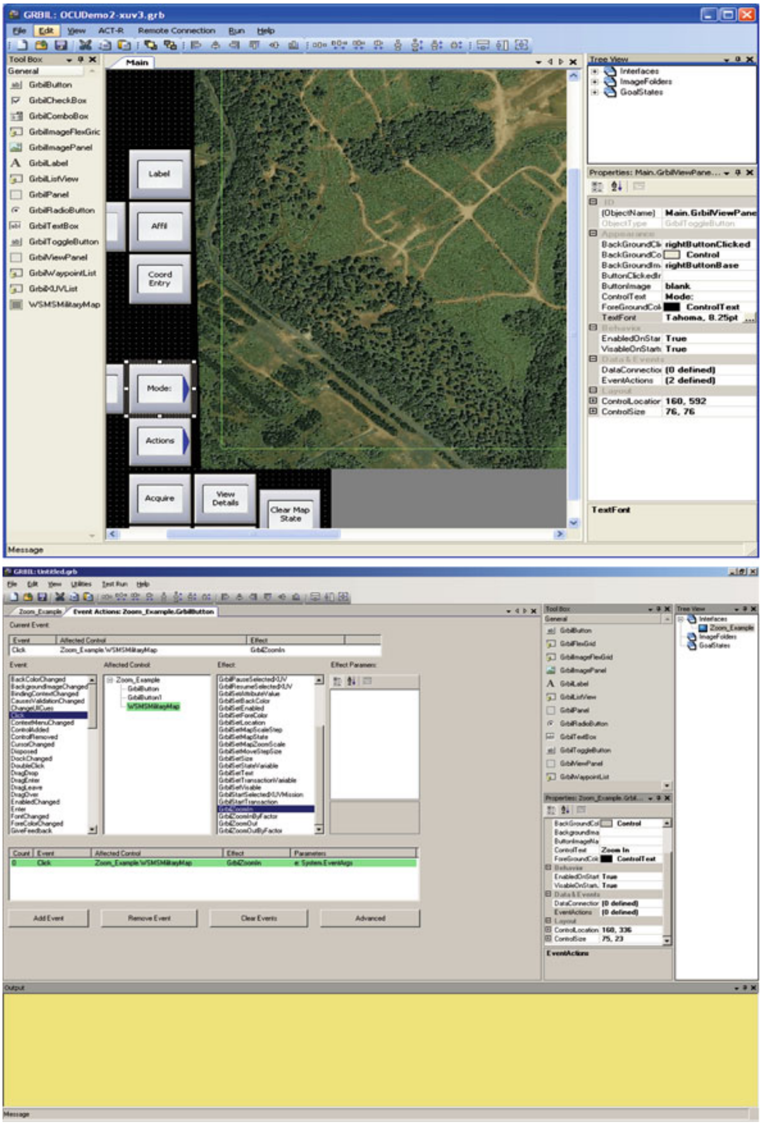


Fig. 3.3 Construction of OCU interface using GRBIL, and assignment of control functionality

select the desired UAV on the map, press the Mode button, press the Add Point button, click on desired waypoints on the map, and then finally press the Execute Plan button. Decision making skills are included by specifying stimulus triggers. Both the sequence of interface operations and the stimulus triggers are automatically compiled into condition-action rules that can be used by the underlying production-engine of the ACT-R cognitive architecture. The operator model can then make human performance predictions. In order to begin validating GRBILs

Fig. 3.4 An interactive mock-up of a dismounted Tactical Control Unit interface developed using GRBIL

timing predictions, performance data from human operators were collected. Two participants familiar with the OCU interface were asked to set up the routes for two vehicles, which involves using the interface to indicate waypoints on a map that each vehicle should pass through. The model predictions and human data are shown in Fig. 3.6.

3.3.4 Constructive Simulation with Dynamic Environment

With the addition of an IMPRINT model of the environment, predictions of more dynamic operator behavior can be made. The creation of monitors for triggering operator procedures automatically gives the operator models the ability to look for text or spatial stimuli (Matessa et al. 2007; Matessa and Brockett 2007). For example, with the OCU interface the operator model can look for text describing the condition of a vehicle or look for possible vehicle intersection and determine an appropriate response (Fig. 3.7).

In addition, interacting models of multiple operators were developed (one setting up and initiating vehicles, one monitoring) and predicted improved performance over a model of a single operator. This was due to the ability of the monitoring operator to react at the same time the initiating operator was busy with a procedure.

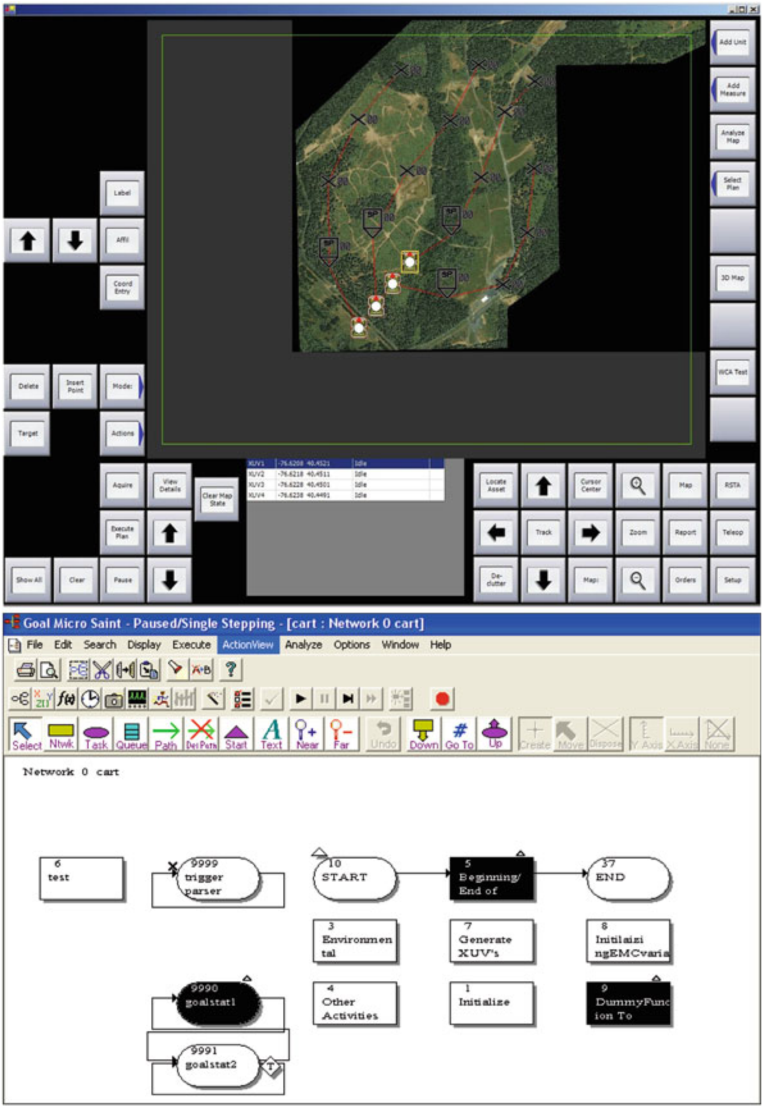


Fig. 3.5 Unmanned vehicles viewed with in the GRBIL mock-up of the OCU interface controlled by an IMPRINT task network model

3.4 Related Work

CogTool (John et al. 2004) is another tool that uses human-in-the-loop modeling by demonstration to address modeling affordability in creating ACT-R models. However, it offers no standard solution for spatial reasoning or integration with dynamic environments (cf. Matessa et al. 2007). Visual processing in CogTool is

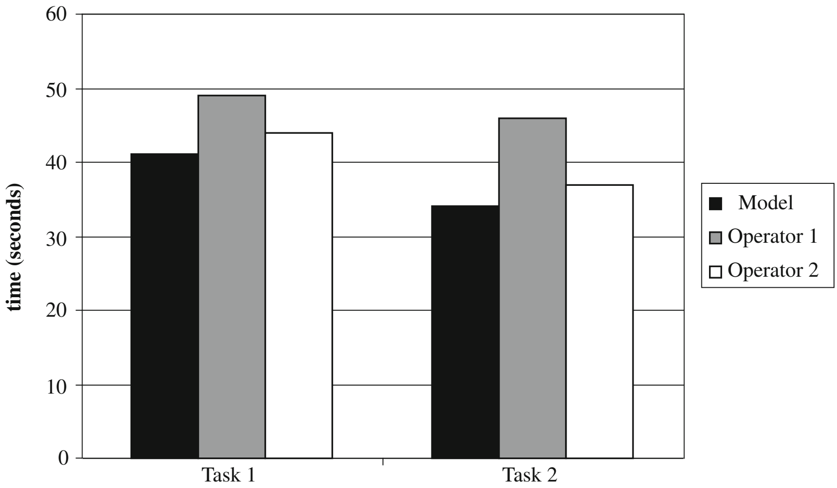


Fig. 3.6 Task completion times for GRBIL model and human operators

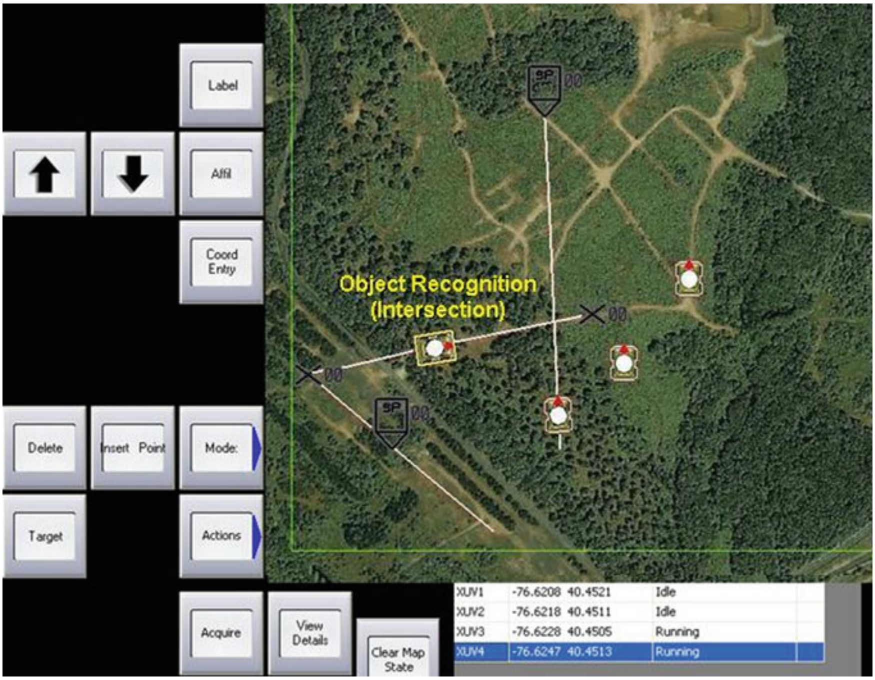


Fig. 3.7 Operator model representation of vehicle intersection

limited to scripted attention movement to interface objects. The environment cannot be monitored to detect a change. The interface used by CogTool is an HTML-based mockup that allows limited transitions between states with each state being represented by an HTML page (although models created with CogTool can later be integrated with dynamic simulators with some hand-coded modifications). Currently, most modeling in architectures (ACT-R: Anderson 2007; Soar: Newell 1990; CPM-GOMS: Vera et al. 2005) do not use interactive modeling by demonstration but rather offline coding.

3.5 Conclusions

The GRBIL tool is intended, in part, to streamline the development of virtual and constructive simulations. The creation of a virtual interface simulation requires only drag-and-drop placement of controls and a menu-driven description of functionality. The creation of a constructive operator simulation requires only human-in-the-loop demonstration of procedures and the creation of stimulus triggers with a simple interface.

While any gains in the efficient development of human performance simulations are welcome, the GRBIL tool also provides a concrete framework in which to explore general questions about human-in-the-loop simulation. Obviously, such simulations are predicated on the assumption that a human is necessary, for whatever reason, to the simulation. But in the case of interface design, we might question that assumption. We previously remarked that a fully constructive simulation of interface use could reveal low-frequency behaviors that might otherwise go unobserved given the relatively fewer "experiments" that can be conducted with human subjects in a virtual simulation.

In addition to that practical concern, there is another, more theoretical argument to consider. To the extent that interface design concerns human–computer interface design it coincides with one of the central organizational principles of cognitive modeling, that of capturing cognition as it is manifested and mediated by computer-based interactions. This is the view of cognition that underlies the human information processor model of Card et al. (1983) and it is one of the methodological cornerstones of modeling within the ACT-R community (cf. Anderson 2007). Of all the domains in which cognitive models have been applied, understanding a human's interaction with a computer interface is one of the most constrained, controlled' and tractable applications. As the GRBIL architecture suggests, the problem at hand is not to model a specific task (e.g., robotic control) but, rather, to model the extent to which the interaction between general cognitive capabilities and a given interface enhances or undermines the performance on such a task (e.g., evaluating a keyboard-based control system).

References

Anderson JR (2007) How can the human mind occur in the physical universe? Oxford University Press, New York

Anderson JR, Bothell D, Lebiere C, Matessa M (1998) An integrated theory of list memory. J Memory Lang 38:341–380

Archer SG, Adkins R (1999) “IMPRINT user’s guide” prepared for US Army Research Laboratory, Human Research and Engineering Directorate

Card S, Moran T, Newell A (1983) The psychology of human–computer interaction. Erlbaum, Hillsdale, NJ

DoD (1998) DoD modeling and simulation (M and S) glossary, DoD 5000. 59-M

John B, Prevas K, Salvucci D, Koedinger K (2004) Predictive human performance modeling made easy. In: Proceedings of CHI, Vienna, Austria, 24–29 April 2004. ACM, New York

Matessa M, Brockett A (2007) Using a diagram representation system with ACT-R. In: Proceedings of sixteenth conference on behavior representation in modeling and simulation. SISO, Baltimore, MD

Matessa M, Mui R (2009) GRBIL: a tool for dynamic operator and environment modeling. In: McDermott P, Allender L (eds) Advanced decision architectures for the warfighter: foundations and technology. Army Research Laboratory, USA, pp 237–254

Matessa M, Archer R, Mui R (2007) Dynamic spatial reasoning capability in a graphical interface evaluation tool. In: Proceedings of the eighth international conference on cognitive modeling. Ann Arbor, MI, pp 55–60

Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge

Vera A, John B, Remington R, Matessa M, Freed M (2005) Automating human-performance modeling at the millisecond level. Human–comput Interact 20(3):225–265

Warwick W, Archer R, Hamilton A, Matessa M, Santamaria A, Chong R, Allender L, Kelley T (2008) Integrating architectures: dovetailing task network and cognitive models. In: Proceedings of the seventeenth conference on behavior representation in modeling and simulation. SISO, Sundance, UT

Chapter 4
An Integrated Pedestrian Behavior Model Based on Extended Decision Field Theory and Social Force Model

Hui Xi, Seungho Lee and Young-Jun Son

Abstract A novel pedestrian behavior model is proposed, which integrates (1) extended decision field theory (EDFT) for tactical level human decision-making, (2) social force model (SFM) to represent physical interactions and congestions among people and the environment, and (3) dynamic planning algorithm involving AND/OR graphs. Furthermore, SFM is enhanced with the vision of each individual, and both individual and group behaviors are considered. The proposed model is illustrated and demonstrated with a shopping mall scenario (a typical mall in the city of Tucson, AZ). Literature survey and observations have been conducted at the mall for data collection and partial validation of the proposed model. The computational environment for human-in-the-loop experiment is also conceptually developed, which will be used to collect more human data in the future. We then developed a simulation model of the considered mall using AnyLogic® software, where each individual in the simulation executes a planning algorithm to select a destination, EDFT for choosing a direction, and extended social force model (ESFM) to adjust its velocity. Using the constructed crowd simulation model, several experiments have been conducted to test the impact of various factors (e.g. consideration of human's vision, group shopping behavior, arrangement of stores, complexity of the model) on several metrics such as the average distance among neighboring shoppers, the movement speed of pedestrians, profit of the shopping mall, and scalability.

H. Xi · Y.-J. Son (✉)
Systems and Industrial Engineering Department,
The University of Arizona, Tucson, USA
e-mail: son@sie.arizona.edu

S. Lee
Mechatronics and Manufacturing Technology Center,
Samsung Electronics Co., Ltd., Suwon-City, Republic of Korea

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_4,

4.1 Introduction

Human crowd dynamics is an essential factor in designing facilities involving a large crowd considering both emergency conditions (e.g. emergency evacuation from a stadium) (Helbing et al. 2005) as well as normal conditions (e.g. shopping mall) (Parisi et al. 2009). Over the past decade, several models have been developed to analyze the underlying mechanism of large-scale crowd behaviors. Xia et al. (2009) classified those models into two major categories: (1) macroscopic models focusing on extremely large crowds whose crowd behaviors are represented via a continuous flow as a whole (as opposed to individualized behaviors) (Gaskell and Benewick 1987; Xia et al. 2009) and (2) microscopic models for studying relatively small crowds whose behaviors emerge from interactions among individuals [e.g. cellular automaton model (Blue and Adler 2001); social force model (SFM) (Helbing et al. 2000); and lattice-gas model (Muramatsu et al. 1999)]. As macroscopic models focus on the continuous flow of crowd as opposed to highly variant, individualized behaviors, they have been mostly applied to the crowd behaviors under competitive situations (e.g. emergency evacuation from a building resulting in a highly dense crowd), where panicking individuals are usually driven by their instincts before every movement and tend to show maladaptive and relentless mass behavior (Helbing 2000). On the other hand, microscopic models pay more attention to individual differences (e.g. preferences, destinations, and tightness in schedule). As our interest in this work is on pedestrian behaviors in a shopping mall (under a normal situation), we will focus on microscopic models.

The SFM introduced by Helbing (2000) is a widely used microscopic model, used for various applications, such as prediction and analysis of congestion, assessment of building or urban layouts and planning of evacuation strategies (Helbing 2005; Moussaïd et al. 2009). Since the original SFM, several researchers have proposed variations or an extended version of it. For example, Hu et al. (2009) extended the model by taking into account the anisotropic characteristic of pedestrian movement in terms of pedestrian vision. Similarly, Parisi et al. (2009) applied the concept of respect area to the original model, which enabled to reproduce the experimental data (e.g. specific flow rates and fundamental diagram of pedestrian flows) for normal conditions.

While extensive works have been performed to enhance the original SFM with various other aspects, limited research works are available in the literature that integrate the human decision-making aspect with SFMs. This has motivated our research, the goal of which is to develop a crowd behavior modelthat integrates (1) tactical level human decision-making, (2) operational-level congestions among people, and (3) detailed-level perceptions (e.g. vision) of individuals. In particular, in the proposed crowd behavior model, decisions on selecting one from alternatives (e.g. destinations and movement directions) are made based on the extended decision field theory (EDFT; Lee et al. 2008), and the physical interactions are represented by the extended social force model (ESFM), which is proposed in this research enhanced with the vision of each person. In addition, pedestrian group behaviors as well as their communications are also explicitly considered in this work. The proposed model is illustrated and demonstrated with a shopping mall scenario providing us with various environmental conditions (e.g. different kinds of shops, obstacles, promotions on the shops) and population variations (e.g. gender, age, preference, schedule, and grouping). Consideration of a rich set of attributes for the environment as well as people will allow us to mimic a real shopping mall environment closely. In particular, the scenario has been built based on the shopping corridor of Tucson Mall (the largest mall in the city of Tucson, AZ). To this end, we have developed a simulation model of the considered shopping mall using AnyLogic® software, where each individual in the simulation executes (1) EDFT (see Sect. 4.2.2), (2) ESFM (see Sect. 4.2.1), and (3) dynamic shopping planning (see Sect. 4.3.4). Using the constructed crowd simulation model, several experiments have been conducted for various purposes, such as (1) to test the impact of the consideration of human vision into SFM on the average distance among neighboring shoppers and the movement speed of pedestrians, (2) test the impact of the number of planned and unplanned shoppers on the profit of the considered shopping mall under low and high density cases, (3) test the impact of group shopping behavior on the profit of the considered shopping mall, (4) test the impact of arrangement of stores in the considered shopping mall on their profit score, and (5) demonstrate the scalability of the proposed model for complex scenarios. Observations have been conducted at Tucson Mall for partial validation of the proposed model and simulations.

The remainder of this paper is organized as follows. In Sect. 4.2, we describe the proposed pedestrian behavior model, its submodules, and techniques employed for the submodules. Section 4.3 describes the development of crowd simulation models based on the proposed behavior model, and computational environments for human-in-the-loop experiments. Section 4.4 discusses five experiments conducted using the developed crowd simulation models. Finally, Sects. 4.5 and 4.6 discuss the conclusions and future work.

4.2 Proposed Integrated Pedestrian Behavior Model

The proposed pedestrian behavior model is based on the integration of extended Decision-Field-Theory (for tactical-level decisions such as selecting a destination or a movement direction during shopping) and extended SFM for dynamic congestions among shoppers and the environment (e.g. walls and obstacles). Each of them is discussed in detail below.

4.2.1 Extended Social Force Model

Helbing (2000) has proposed an SFM, where the motives and impacts to a pedestrian crowd are represented by a combination of physical and psychological forces (which are translated into the acceleration equation). Equation 4.1 depicts the formulation of changing velocity at time t, where a pedestrian i's velocity $\boldsymbol{v}_i$ is determined by his/her desired speed $v_i^0(t)$ and desired direction $e_i^0(t)$ as well as interactions with other individuals and obstacles.

$$m_i \frac{dv_i}{dt} = m_i \frac{v_i^0(t) e_i^0(t) - v_i(t)}{\tau_i} + \sum_{j(\neq i)} f_{ij} + \sum_{W} f_{iW} \tag{4.1}$$

where m is the pedestrian mass, τ_i is a time constant related to the relaxation time of the particle to achieve v_i.

The first term on the right-hand side of Eq. 4.1 represents the impact from the pedestrian's self-consciousness while f_{ij} and f_{iW} illustrate interaction forces from pedestrian j and the wall W, respectively. The pedestrians try to keep a velocity-dependent distance from other people and the walls so as to construct a comfortable zone for themselves. The interaction force consists of a socio-psychological force f_{ij}^{psy} resulting from the distance between each other, and a physical force f_{ij}^{phy} inspired by counteracting body compression and sliding friction. The total force exerted by pedestrian j to pedestrian i is calculated as below:

$$f_{ij} = f_{ij}^{\mathrm{psy}} + f_{ij}^{\mathrm{phy}} \tag{4.2}$$

$$f_{ij}^{\mathrm{psy}} = A_i \exp\left(\frac{r_{ij} - d_{ij}}{B_i}\right) \boldsymbol{n}_{ij} \tag{4.3}$$

where A and B are constants that describe the strength and range of psychological interaction, r_{ij} is the sum of radii of pedestrian i and j, dij is the distance between i and j, $\boldsymbol{n}_{ij}$ is the unit vector pointing from j to i.

$$f_{ij}^{\mathrm{phy}} = \mathrm{k}g(r_{ij} - d_{ij})\boldsymbol{n}_{ij} + \kappa g(r_{ij} - d_{ij})\Delta v_{ji}^t \boldsymbol{t}_{ij} \tag{4.4}$$

where k and κ are the normal and tangential elastic restorative constants, $\boldsymbol{t}_{ij}$ is tangential unit vector perpendicular to $\boldsymbol{n}_{ij}$, v_{ij}^t is the tangential projection of the relative velocity seen from pedestrian j ($\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$), and g is 1 if $d_{ij} > r_{ij}$ and 0 otherwise.

While the original SFM (Helbing 2000) discussed above has been extensively applied to pedestrian behavior modeling, there exist two improvement opportunities when applying to a real-life human behavior. First, the original SFC computes a force impact between every pair of agents in the environment. In other words, there will be a force even between agents who are significantly far away from each other, which is not realistic. Second, the social force between agents is always positive implying that all agents are psychologically against each other. However, this is not the case for friends or family members, who usually stay close to each other while moving (shopping in our case) under a nonemergency condition. To address these two problems, we extended the original SFC. Details of each modification will be discussed in Sects. 4.2.1.2 and 4.2.1.3.

4.2.1.1 Connection Range Impact on Social Force Model

Experimental investigations have demonstrated the self-organization phenomenon to be a typical characteristic of pedestrians. With the help of technologies like video tracing, researchers have further found that self-organization is caused by collision avoidance behavior. In other words, pedestrians tend to keep a suitable distance from others to avoid bumping into one another (Ma et al. 2010). Therefore, in this work we define a connection range, CR, for each agent in the environment. Before applying the force (see Eq. 4.2) between two agents, ESFM will evaluate the distance, d_{ij}, between them first and compare it with the connection range. Only if $d_{ij} < \text{CR}$, are these two agents connected and have a force affecting their movement (see Eq. 4.5). However, there is an exception for the group members, which will be discussed in Sect. 4.4.1. Considering the radius of agent r_i in the range between 0.25 and 0.35 m (Helbing et al. 2000), we have chosen CR as 5 m in this work so that pedestrians could get more information from their surroundings (Ma et al. 2010).

$$d_{ij}\begin{cases} > \text{CR agent } i \text{ and } j \text{ are not connected} \\ \leq \text{CR agent } i \text{ and } j \text{ are connected} \end{cases} \tag{4.5}$$

4.2.1.2 Psychological Attraction Between Group Members

Pedestrian populations in the shopping mall (case study in this research) can be categorized into two types: individual shoppers and group shoppers (see Sect. 4.3 for more details about the considered scenario). Among individual shoppers or shoppers from different groups, a psychological force in the original SFC is applicable to keep a comfort distance between them. However, for shoppers belonging to the same group, the psychological force will prohibit them from staying close to each other. Therefore, we propose a modification of the psychological force for members of the same group (see Sect. 4.4.1 for more details), where an intimate factor I_{ij} (see Eq. 4.6) is multiplied with the psychological force (see Eq. 4.3). The main idea is that a positive psychological force is applicable for people belonging to different groups while a negative psychological force is applicable for people belonging to the same group (Helbing 2005).

$$I_{ij} = \begin{cases} 1, & i \text{ and } j \text{ are group members} \\ -1, & \text{otherwise} \end{cases} \tag{4.6}$$

4.2.1.3 Pedestrians' Reactions According to Their Visions

In the original SFM, obstacles located at the same distance (without considering the concept of vision or sight) from a pedestrian enforce the equal psychological force on him/her. In a real shopping environment, however, people usually pay

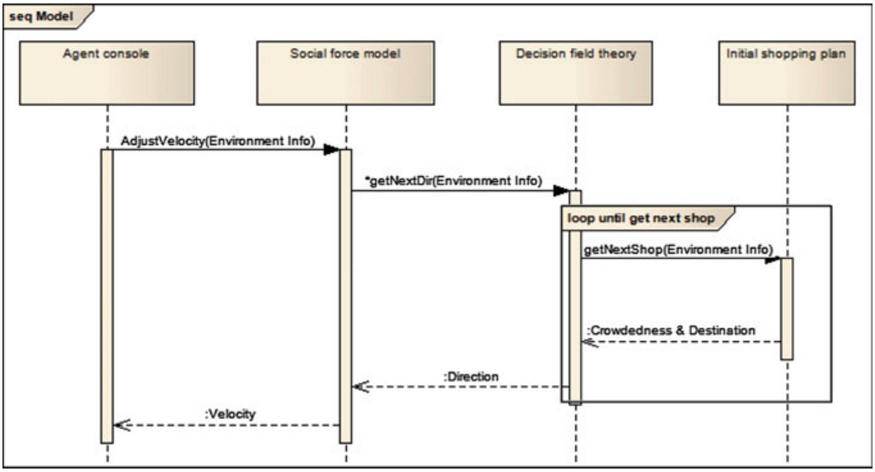


Fig. 4.1 Sequence diagram of components of the proposed pedestrian behavior model

more attention to objects within their vision than to those out of their sight. To resolve this problem, we incorporate this concept of vision by defining a visible area for each agent in this work. From the view of an agent, only neighbors in his visible area could affect his movement with psychological force. Neighbors behind his back (out of vision) may provide influence only via a physical force (e.g. contact). In our proposed model, a visible area (range) is defined with a half circle in front of each pedestrian (±90 degree angle from the pedestrian's current moving direction) (see Figs. 4.1, 4.2). The vision formula is given in Eq. 4.7, where $\varphi_{ij}(t)$ is the angle between direction $\boldsymbol{e}_i(t)$ and normalized vector $\boldsymbol{n}_{ij}(t)$. With$\varphi_{ij}(t) > 90°$, cos $(\varphi_{ij}(t))$ (<0) is rounded up to 0, while$\varphi_{ij} < 90°$ will round $\cos(\varphi_{ij}(t))$ ($\geq$0) up to 1. Based on this visible area, a modified social force exerted from pedestrian j on pedestrian i is given in Eq. 4.8.

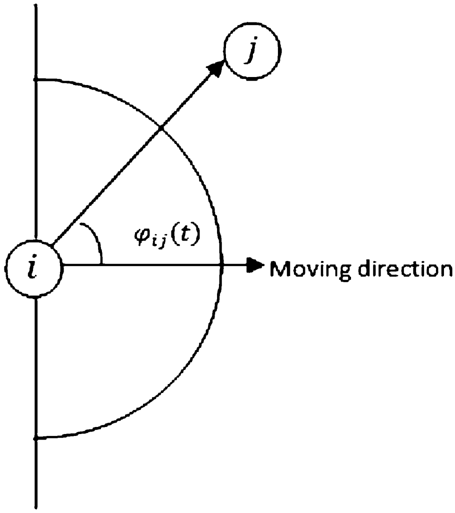
$$\cos(\varphi_{ij}(t)) = \frac{\boldsymbol{n}_{ij}(t)\boldsymbol{e}_i(t)}{\|\boldsymbol{n}_{ij}(t)\|\|\boldsymbol{e}_i(t)\|} \tag{4.7}$$

$$f_{ij}^{\mathrm{psy}} = A_i \exp\left(\frac{r_{ij} - d_{ij}}{B_i}\right)\boldsymbol{n}_{ij} I_{ij} \lceil \cos(\varphi_{ij}(t)) \rceil \tag{4.8}$$

4.2.2 Incorporating EDFT into the Pedestrian Model

It is generally agreed that decision making about walking trips takes place simultaneously at two or more levels: (1) decisions about basic strategy of the trip, (2) route choice, and (3) local spatial behavior considering velocity, trajectory,

Fig. 4.2 Visible area for each agent



stops, and attention direction (Zacharias 2005). In this work, it is assumed that a basic strategy (which shops to stop by) and a route choice (in what sequence) are fixed for each individual. Therefore, pedestrian's decisions, of interest to us, are focused on changing their movement directions. Since pedestrians adjust their actual direction from time to time due to the interaction force, the EDFT is employed in this work to mimic this dynamic human decision deliberation process.

Decision Field Theory (DFT) is a psychology-based model and has been widely used for mimicking human deliberation process in making decisions under uncertainty (Busemeyer and Diederich 2002; Busemeyer and Townsend 1993). Lee et al. (2008) extended the original DFT to cope with a dynamically changing environment, where a Bayesian Belief Network (BBN) was used to infer human decision attributes under the dynamically changing environment. In the shopping mall scenario considered in this research, the environmental conditions (e.g. crowd density and destinations) change dynamically for individuals. Therefore, we integrate the EDFT into our proposed pedestrian model to better mimic pedestrians' deliberation on direction changes. Our EDFT is able to model (1) the change of evaluation on the options and (2) the change of human attention along with the dynamically changing environment. The formulation of EDFT is given in Eq. 4.9, which illustrates the dynamic evolution of preferences P among options during the deliberation time h.

$$P(t+h) = \mathrm{SP}(t) + \mathrm{CM}(t+h)W(t+h) \tag{4.9}$$

In our work, pedestrians change their movement direction according to the environment around them, for example, the increase/decrease of crowd density or the position of their next destination. Definitions of the main elements of EDFT are explained below:

- $M(t)$ is the value matrix ($n \times m$ matrix, where each n option has m attributes) representing the subjective perceptions of a decision-maker by $M(i, j)$. In our

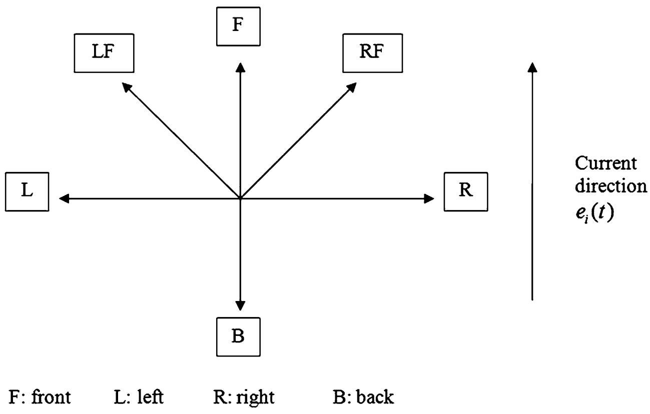


Fig. 4.3 Potential directions for each decision making

case (choosing a direction), a 6 × 2 matrix (see Eq. 4.10) is used, where pedestrians have six options (see Fig. 4.3), and each direction corresponds with two attributes (crowd density and destination) that affect their choice. If the next destination is within direction i, the entry value $\mathrm{Des}_i(t)$ is 0.5; otherwise, it is 0.1. To decide whether a destination is within a particular direction, $\boldsymbol{n}_{\mathrm{iD}}$ to denote a vector from pedestrian i to the destination. If the angle between $\boldsymbol{n}_{\mathrm{iD}}$ and the direction is ≤22.5°, we claim that the destination is in this direction. Thus, the value matrix M has a dynamic representation as shown in Eq. 4.10, and its values change whenever the underlying conditions change.

$$M(t) = \begin{bmatrix} \mathrm{Den}_F(t) & \mathrm{Des}_F(t) \\ \mathrm{Den}_{LF}(t) & \mathrm{Des}_{LF}(t) \\ \mathrm{Den}_{RF}(t) & \mathrm{Des}_{RF}(t) \\ \mathrm{Den}_L(t) & \mathrm{Des}_L(t) \\ \mathrm{Den}_R(t) & \mathrm{Des}_R(t) \\ \mathrm{Den}_B(t) & \mathrm{Des}_B(t) \end{bmatrix} \tag{4.10}$$

where

$$\mathrm{Des}_i(t) = \begin{cases} 0.5, & \text{next destination is in direction } i \\ 0.1\,, & \text{otherwise} \end{cases} \tag{4.11}$$

$$\mathrm{Den}_i(t) = \begin{cases} 0.2, & \text{if } \mathrm{cd}_i(t) < 5 \\ 0.4, & \text{if } 5 \le \mathrm{cd}_i(t) < 15 \\ 0.6, & \text{otherwise} \end{cases} \tag{4.12}$$

W(t) is a weight vector allocating the portion of human attention to each column j (attribute) of M through $W(j, 1)$, which is the only dynamically changing element in the original DFT (Busemeyer and Townsend 1993). In the shopping mall environment, it is assumed that pedestrians intend to arrive at their destination as soon as possible. However, when the environment is really crowded, they tend to put more weight on the impact of crowd density. Equations 4.12 and 4.13 depict *W(t)* used in the considered shopping mall scenario, where $\mathrm{cd}_i(t)$ denotes the crowd density.

$$W(1,1) = \begin{cases} [0.2, 0.3]^T, & \text{if } \mathrm{cd}_i(t) < 5 \\ [0.25, 0.6]^T, & \text{if } 5 \le \mathrm{cd}_i(t) < 15 \\ [0.4, 0.6]^T, & \text{otherwise} \end{cases} \tag{4.13}$$

$$W(1,2) = 1 - W(1,1) \tag{4.14}$$

- S demonstrates the stability of preference to each option by its structure. The diagonal elements of S represent the memory from the previous preference state while off-diagonal elements give the inhibitory interactions among competing options. Here, it is assumed that the same amounts of memory and interaction effects are given to the options: (1) matrix S is assumed to be symmetric and (2) diagonal elements of S are assumed to have the same value. Moreover, all eigenvalues λ_i of S are >1 in magnitude to make the linear system stable ($|\lambda_i| < 1$). Besides, from Fig. 4.3, we can see larger interactions between directions within 45° than those in 90 or larger degrees. Considering this, we have defined S matrix (see Eq. 4.15).
- C is the contrast matrix comparing the weighted evaluation of each option, *MW(t)*. In our case, each option is evaluated independently, thus C tends to be I (identity matrix). Given the aforementioned elements and our six-option scenario, the corresponding DFT formula, as defined in Eq. 4.9, is described in Eq. 4.15.

$$\begin{pmatrix} p_1(t+h) \\ p_2(t+h) \\ p_3(t+h) \\ p_4(t+h) \\ p_5(t+h) \\ p_6(t+h) \end{pmatrix} = \begin{pmatrix} 0.9 & -0.6 & -0.6 & -0.4 & -0.4 & -0.1 \\ -0.6 & 0.9 & -0.4 & -0.6 & -0.2 & -0.6 \\ -0.6 & -0.4 & 0.9 & -0.2 & -0.2 & -0.2 \\ -0.4 & -0.6 & -0.2 & 0.9 & -0.1 & -0.4 \\ -0.4 & -0.2 & -0.2 & -0.1 & 0.9 & -0.4 \\ -0.1 & -0.6 & -0.2 & -0.4 & -0.4 & 0.9 \end{pmatrix} \begin{pmatrix} p_1(t) \\ p_2(t) \\ p_3(t) \\ p_4(t) \\ p_5(t) \\ p_6(t) \end{pmatrix}$$
$$+ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} M_{11}(t+h) & M_{12}(t+h) \\ M_{21}(t+h) & M_{22}(t+h) \\ M_{31}(t+h) & M_{32}(t+h) \\ M_{41}(t+h) & M_{42}(t+h) \\ M_{51}(t+h) & M_{52}(t+h) \\ M_{61}(t+h) & M_{62}(t+h) \end{pmatrix} \begin{pmatrix} W_{11}(t+h) \\ W_{21}(t+h) \end{pmatrix} \tag{4.15}$$

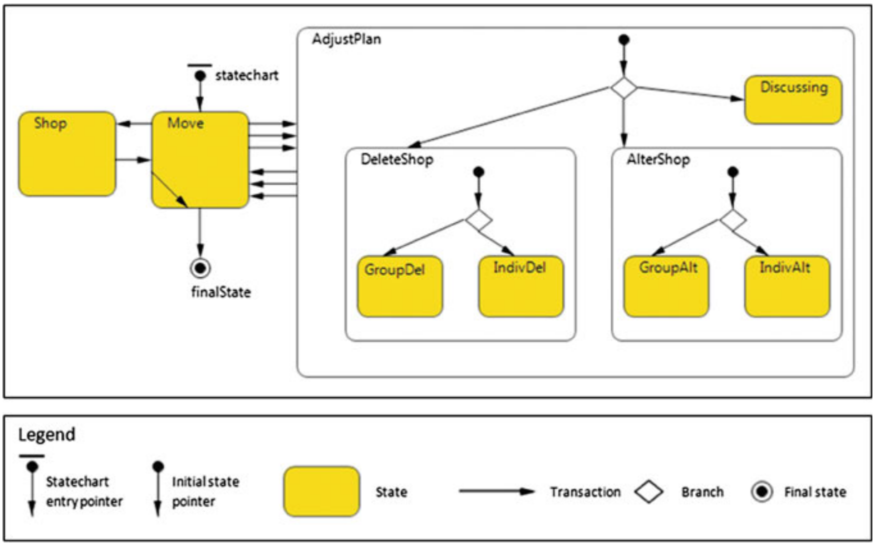


Fig. 4.4 State charts for the shopper's behavior

4.3 Development of Agent-Based Simulation Based on Proposed Pedestrian Model

This section describes the development of a crowd simulation model for a shopping mall scenario, where behaviors of individual shoppers are based on the proposed, integrated pedestrian behavior model (see Sect. 4.2). We employed a two-layer modeling principle (Hamgami and Hirata 2003) in the development of the crowd simulation model to reduce the complexity of the modeling process, where agents and the environment they interact with are modeled separately in two conceptual layers. The interactions between agents and the environment are analogous to how humans behave in the real world. Agents evaluate the surroundings and try to make optimal decisions so as to achieve their intentions. Figure 4.4 depicts a state chart for shoppers, which contains different states in which shoppers will be in and their transitions. More details about each state and simulation models will be discussed in the following sections.

4.3.1 Shopping Mall (Case Study) and Customer Classifications

A shopping mall scenario has been designed, and its simulation implemented using AnyLogic® 6.4 agent-based simulation software. The considered scenario covers eight shops of four different types: three clothing shops, two sports shops, two

Table 4.1 Shop list in the simulation scenario

Shop ID	Shop type	Shop ID	Shop type
A, B, G	Clothes shop	C	Candy shop
D, F	Sports shop	E, H	Beauty shop

Table 4.2 Agent categorizations

Shopping style		Agent type		
Unplanned		Female adult	Male adult	
Planned	Group	Female adult	Male adult	Child
Planned	Individual	Female adult	Male adult	

beauty shops, and one candy shop. Each shop has its own ID as listed in Table 4.1. Each type of shop has its target customers. For instance, female customers may be more interested in beauty shops while males may be more interested in sports shops. In this work, to enhance the validity and realism of the constructed simulation model, we have categorized customers in multiple ways. Table 4.2 depicts multiple categories of customers considered in this work. First, customers are tagged with three agent types based on their gender and age: (1) female adult, (2) male adult, and (3) child. In this work, the agent type is based on the ratios of sex and age of the Tucson population (http://www.maps-n-stats.com/). Therefore, agent type is determined based on the discrete, empirical statistical distribution shown in Eq. 4.16.

$$P_{agent\ type} = \begin{cases} [0, 0.4], & \text{agent type is female adult} \\ (0.4, 0.8], & \text{agent type is male adult} \\ (0.8, 1.0], & \text{agent type is child} \end{cases} \tag{4.16}$$

Besides utilitarian-oriented shopping, there also exists window shopping orientation and recreational shopping. Therefore, based on their shopping style, customers are categorized into planned shoppers (people who go to the mall with specific shopping plan) and unplanned shoppers (who do not have a specific shopping plan). Upon arriving at the mall, planned shoppers already have in mind which shops they will visit. Planned shoppers are further partitioned into group shoppers (those who do shopping with friends or family members) and individual shoppers. In this work, it is assumed that all the unplanned shoppers are individual shoppers. By combining agent type with other categorizations (e.g. planned vs. unplanned shoppers; individual verses group shoppers), we can enhance the flexibility of pedestrians' behaviors as well as their adaptability to the environment.

4.3.2 Algorithm for Movement of Pedestrians

In this section, the algorithm for movement of pedestrians is discussed in detail. Figure 4.5 depicts a flowchart of the movement algorithm. As discussed in Sect. 4.2.2, a desired destination is used as part of input M for EDFT during the

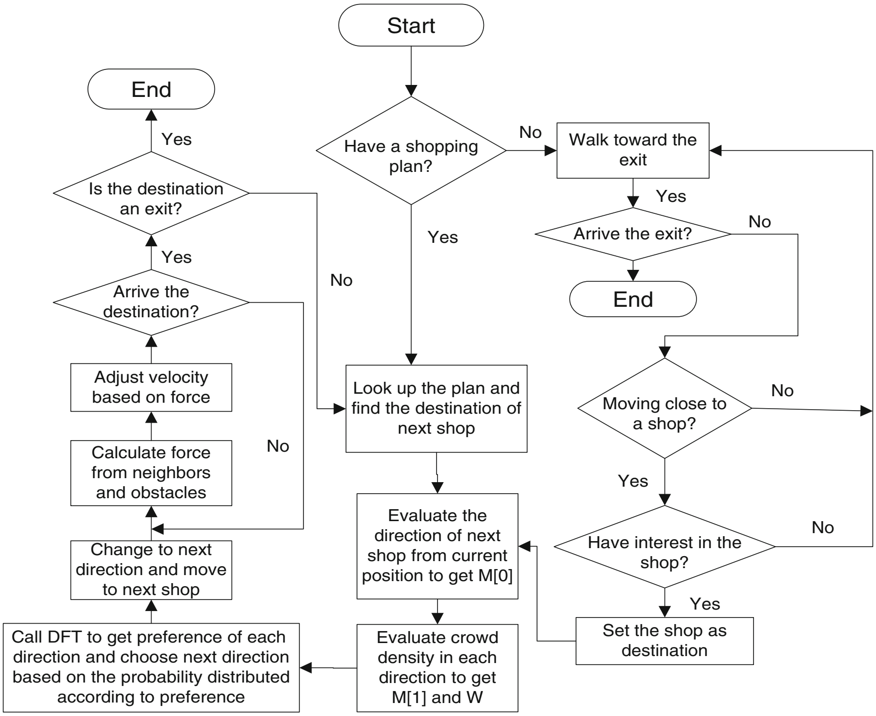


Fig. 4.5 Pedestrian moving algorithm

decision deliberation on directions. For the planned shoppers, a potential destination is obtained from their shopping plans. On the other side, unplanned shoppers normally set the closest shop as their potential destination. In this work, an attribute (crowdedness threshold) is defined to represent the largest number of people that each shopper could accept to shop with in the same store. Before a shopper enters his/her planned destination (shop), he/she will evaluate it based on several criteria such as their interest level, shop's attraction level, and crowdedness level, and confirm the desired destination based on their evaluations. More details about evaluation of the destination will be illustrated in Sects. 4.3.3 and 4.3.4 for unplanned and planned shoppers, respectively. As soon as an agent (shopper) comes up with a desired destination, he/she utilizes EDFT to determine their next moving direction. To implement/compute EDFT, a Java Matrix Package (JAMA) has been embedded into our simulator (Anylogic model). Then, by calculating the physical and social forces based on the surroundings along the moving direction, each agent adjusts its velocity in SFM (see Eq. 4.1).

4.3.3 Destination Confirmation Algorithm for Unplanned Shoppers

As mentioned earlier, unplanned shoppers wander around the shopping mall, without knowing in advance which shop they will visit. When they pass by a shop, they will set it as a potential destination if the shop's crowdedness level is below their threshold. Then, they evaluate the shop based on their personal interest and the shop's attraction level by Eq. 4.17. Based on this evaluation, the potential destination may become a confirmed destination. For instance, a male, unplanned shopper who is more interested in sports shoes than cosmetics may enter a sports shop but not a beauty shop when he passes by one of them. However, if the beauty shop has a special promotion (e.g. big sale), he may still visit the shop. Equation 4.17 depicts a probability function on whether unplanned shoppers enter a shop or not.

$$P_{iw} = \alpha I_{iw} + \beta A_{iw} \tag{4.17}$$

I_{iw} denotes the interest level of agent i for shop w, while A_{iw} describes the attraction level of shop w towards agent i. Constants α and β are the weight values assigned to the interest level and attraction level, respectively. Both variables (I_{iw}, A_{iw}) and constants (α, β) range from 0 to 1. In our model, we give the same weight to I_{iw} and A_{iw} by setting $\alpha = \beta = 1$. Therefore, in the normalization step, the value obtained for P_{iw} is divided by 2 in order to obtain its normalized value. Here, we assume that $P_{iw} > 0.5$ indicates that agent i is definitely attracted by shop w and will enter this shop.

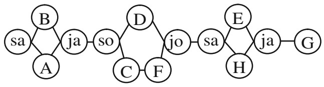
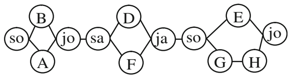
4.3.4 Planning Algorithm for Planned Shoppers

Planned shoppers obtain their potential destinations based on their shopping plan, and evaluate them in the same way as unplanned shoppers (see Sect. 4.3.3). If the potential destination does not meet one of their three criteria (interest level, shop's attraction level, and shop's crowdedness level), planned shoppers will need to decide whether they will skip the shop or move to another shop (of the same kind). This decision process is defined as plan adjustment. The following sections discuss the design of initial shopping plans and the plan adjustment algorithm for planned shoppers in greater detail.

4.3.4.1 Alternative Initial Shopping Plans for Planned Shoppers

When planned agents arrive at the entrance of the mall, they will be offered with initial shopping plans according to their characteristics (see Table 4.2). For individual planned shoppers, the plan is designed based on their agent type.

Table 4.3 Predefined shopping plans for individual shoppers

Agent type	Initial shopping plan
Female adult	
Male adult	

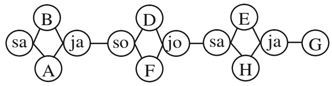
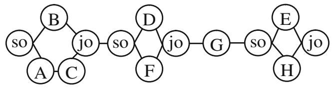
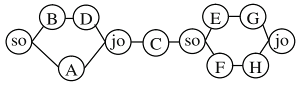
A -- H: shop (See Table 4.1)
so: separate end of OR operation jo: joint end of OR operation
sa: separate end of AND operation ja: joint end of AND operation

For instance, female shoppers may want to visit all three clothes shops and two beauty shops if their schedule permits. Thus their initial shopping plan will include these shops. For a male shopper, a different plan will be designed according to his personal need. Table 4.3 depicts the shopping plans for individual shoppers. Plans in Table 4.3 contain alternatives in stores to visit (using OR junctions) or in the sequence of stores to visit (using AND junctions).

Based on the survey conducted by Kuruvilla et al. (2009) and observations made in the real shopping environment, we have partitioned shopping groups into three types based on their shopping interest and group members' personal characteristics: (1) female groups consisting of female shoppers whose interests mainly focus on beauty and clothing shops; (2) mixed gender groups with both male and female members; and (3) family with-kid groups. Family groups are mixed-gendered groups that need to balance shopping interests, but they will include candy shops on their plan due to kids. For group shoppers, their shopping plans are not based only on one person's interests, but should consider the need of all the members in the group and achieve a balance for the whole group's interests. Taking a mixed-gender group for example, while female members may need to visit more beauty shops and males may need to visit sports shops, the group shopping plan would include both types but only one shop (less than what is preferred by each party) for each type. As another example, if there is a kid in the group (family group), adult members may have to give up one of their shops of interest (e.g. sports, clothes, or beauty) since they may need to go to the candy shop with the kid. When group shoppers enter the mall, they will be assigned with a group ID ($0 \sim 8$), which will indicate the group type they belong to. Table 4.4 depicts group types and the corresponding initial plans. The group frequencies vary by scenario (e.g. different ethnicity). For example, about 70% of Indian people always shop with families while the percentage is lower for American shoppers (Kuruvilla et al. 2009).

As shown in Tables 4.3 and 4.4, initial shop plans contain alternatives, and one of them will need to be selected based on the current situation. Taking a female adult as an example in Table 4.3, she needs to visit shop A and shop B upon her

Table 4.4 Predefined shopping plans for group shoppers

Group ID	Agent type	Initial shopping plan
0, 1, 5	Female group	
2, 4, 8	Mix-gender group	
3, 6, 7	Family with kids group	

arrival, but in any sequence. This selection is based on both her current position and the shop selection probability. If she enters the mall at area 1 in Fig. 4.8, the probability of selecting shop B is higher than that of shop A because of its proximity. In this work, we use P_{sw} to denote the probability that shop w will be selected by a nearby agent and Pos_k to denote the position of agent. Then the probability of selecting shop A and shop B as the first destination is given by Eqs. 4.18 , and 4.19, respectively.

$$P_A = \mathrm{Pos}_2 * P_{sA} + \mathrm{Pos}_1 * (1 - P_{sB}) \tag{4.18}$$

$$P_B = \mathrm{Pos}_1 * P_{sB} + \mathrm{Pos}_2 * (1 - P_{sA}) \tag{4.19}$$

where

$$\mathrm{Pos}_k = \begin{cases} 1, & \text{if an agent's entering position in area } i; \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

$$P_{sA} = P_{sB} = 0.7 \tag{4.21}$$

4.3.4.2 Plan Adjustment for Individual Planned Shop

As described in Sect. 4.3.4, a potential destination of planned shoppers becomes a confirmed one if the shop's crowdedness level is below their threshold. If a considered shop is too crowded, they may want to skip it and go to a next planned one or adjust their plan to visit a different shop (of the same kind). Figure 4.6 illustrates the procedure in which planned shoppers adjust their shopping plans according to dynamically changing surroundings. For individual shoppers, they evaluate a shop according to Eq. 4.17 just as unplanned shoppers do. We use P_{iw} to denote the probability that agent i would like to stop by shop w. If P_{iw} is larger than 0.5, they will choose to visit a similar shop instead. Otherwise, they will skip the shop and set a next planned shop as the potential destination.

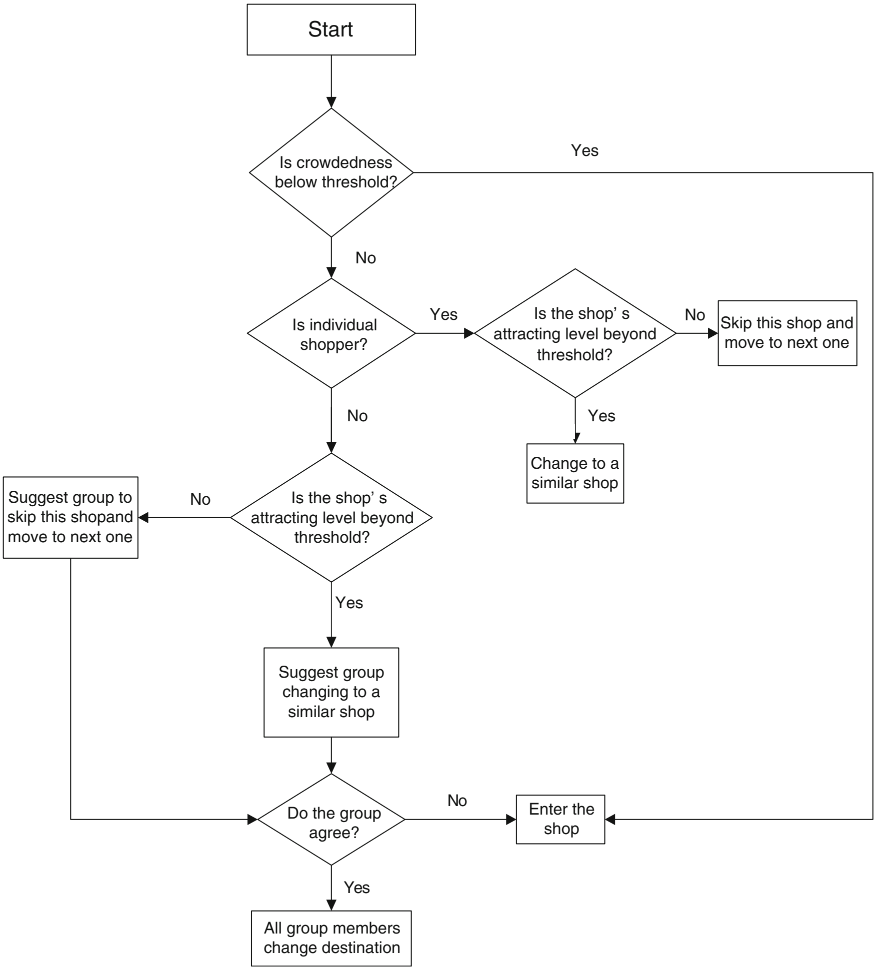


Fig. 4.6 Plan adjustment algorithm against dynamic situations

4.4 Experiments and Results

Using the crowd simulation model constructed based on the proposed pedestrian model and data (survey and observations), we have conducted several experiments for various purposes, such as (1) to test the impact of consideration of human's vision into SFM on the average distance among neighboring shoppers and the movement speed of pedestrians, (2) test the impact of the number of planned and unplanned shoppers on the profit of the considered shopping mall under low and high density cases, (3) test the impact of group shopping behavior on the profit of the considered shopping mall, and (4) test the impact of arrangement of stores in the considered shopping mall on their profit score. Also, we tested the scalability of the proposed model by increasing the number of agents in the simulation. The detailed design of each experiment, results, and analyses are described in the following sections.

4.4.1 Significance of Consideration of Vision in Social Force Model

The goal of this experiment is to test the significance of consideration of human's vision into SFM, which is part of the proposed pedestrian behavior model in this work. As mentioned in Sect. 4.2, one of the group characteristics is the positive social force among the group members. Unlike individual and group shoppers belonging to different groups, group members of the same group stay close to each other and move together. In Sect. 4.2.1.1, a concept of connection range (CR) was discussed for the social force between agents except the group members. The exception for the group members is that even if two group members are out of their connection range, there is still a psychological force f^{psy} between them in order to reduce the distance between them. Once they get closer and are within the connection range, a physical force f^{phy} begins working to avoid any friction or collision between them. Figure 4.7 depicts the forces between group members from group member i's view. Equation 4.22 depicts the resultant force function for the group members.

$$f_{ij} = \begin{cases} f_{ij}^{psy} + f_{ij}^{phy}, & d_{ij} \leq CR \\ f_{ij}^{psy}, & d_{ij} > CR \end{cases} \tag{4.22}$$

By considering the human vision in SFM, pedestrians will have a psychological force against only those neighbors in front of them and adjust their speed consequently. In this case, their resistance force is reduced; therefore, they are hypothesized to move faster. Then we do the Student's t test with alternative hypothesis H_1 and null hypothesis H_0 stated as below:

H_1 *Pedestrians will tend to move faster when vision is considered*
H_0 *Pedestrians will not move faster when vision is considered*

Figure 4.8 depicts a snapshot of the shopping mall simulator that we have developed, where 100 pedestrians are moving along the hallway towards the exit. This experiment was designed in a way that pedestrians do not visit any shop. Therefore, it allows us to test the significance of consideration of human vision into SFM in a general case, where the average speed of pedestrians is used as a metric. Experiments have been conducted with 30,100, and 1,000 pedestrians to compare average speeds between models with and without consideration of human vision. Statistics shown in Table 4.5 are based on 16 samples collected every 10 s in each case. By using student's t testing with equal sample sizes and unequal variance, we obtained p-value <0.001 which accepts our hypothesis. The experimental results reveal that our intuition on faster movement of pedestrians when we

Table 4.5 Statistics of Student's t testing on the significance of vision in SFM

Number of pedestrians	$\overline{X_1}$	$\overline{X_2}$	s_1^2	s_2^2	$n_1 = n_2$	t	p
30	11.91306	11.83306	0.00015	0.00041	16	13.522	<0.001
100	11.77788	11.43238	0.00325	0.04586	16	6.236	<0.001
1000	11.95081	11.45106	0.00010	0.00732	16	23.207	<0.001

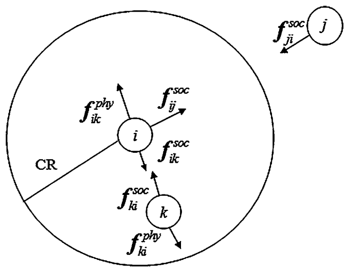


Fig. 4.7 Force execution between group members (from group member i's view)

consider human vision is correct regardless of the crowd density. Therefore, consideration of human vision into SFM has been found to be significant (Fig. 4.9).

4.4.2 Impact of Unplanned Shoppers on the Number of Visits to Shops

The goal of this experiment is to analyze the impact of the number of unplanned shoppers on the number of visits to the shops (and therefore profit of the shopping mall) during the same time period. As mentioned in Sect. 4.3.3, shoppers (planned and unplanned) will evaluate the crowdedness of a shop before entering it. Equation 4.23 depicts the probabilities that planned and unplanned shoppers will purchase items used in this experiment. Equation 4.24 depicts the profit of shop w, where m denotes the minimum crowdedness threshold of the shopper in the shop. If shops are mostly filled with unplanned shoppers, they may lose the opportunity to attract planned shoppers whose probability of purchasing is higher, reducing the profit of the shop. For unplanned shoppers, we assumed equal chances for them to make a purchase or not while visiting a shop. Many previous studies (Zhuang et al. 2006; Babin et al. 1994; Batra and Ahtola 1991; Baumann et al. 1981) found that the buying intention tends to increase shoppers' buying of non-food products such as clothes; we give higher purchase probability to planned shoppers.

$$\mathrm{Pr}_i(\text{purchase})\begin{cases} <0.5, & \text{is a planned shopper} \\ =0.5, & \text{is an unplanned shopper} \end{cases} \tag{4.23}$$

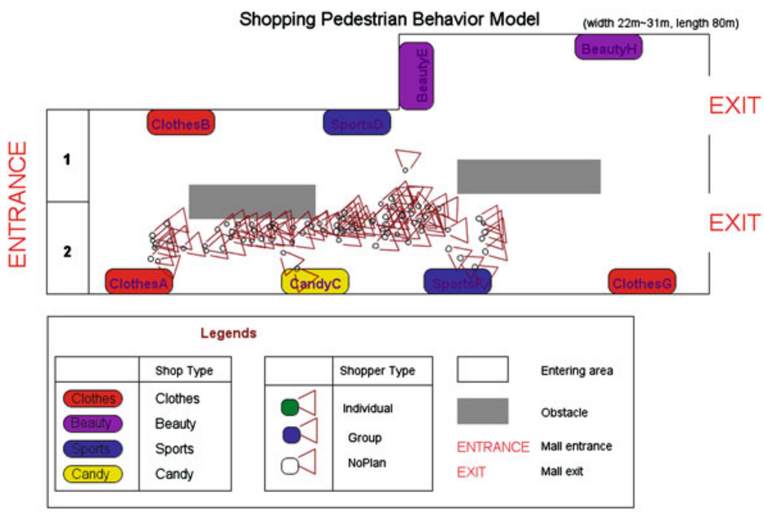


Fig. 4.8 Snapshot of a shopping mall simulation with 100 participants

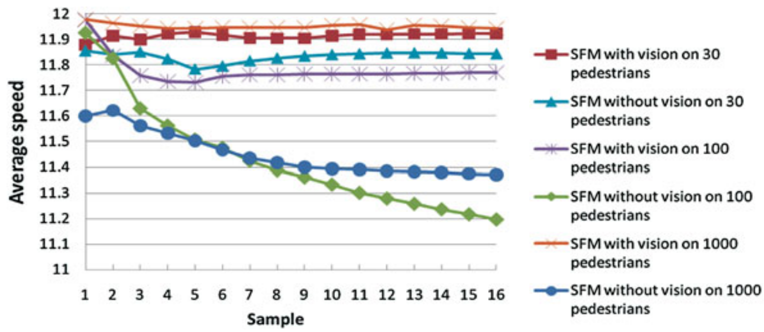


Fig. 4.9 Average speed of pedestrians in SFM with and without considering of vision

$$\text{ProScore}_w = \sum_i^m \Pr_i(\text{purchase}) \tag{4.24}$$

The first experiment has been conducted with a high density envrionment, where the total profit score of the mall for 100 min is compared under three different conditions: (1) 54 planned shoppers and 76 unplanned shoppers in the mall, (2) 57 planned shoppers and 117 unplanned shoppers in the mall, and (3) 96 planned shoppers and 78 unplanned shoppers in the mall. By setting the purchasing probability of planned shoppers as 0.6 and 0.8 repectively, we could see from Figs. 4.10 and 4.11 the impact of shoppers' buying intention on the mall's profit gaining. As depicted in Figs 4.11a and b, cases 1 and 2 indicate that the profit score does not increase greatly when about 40 additional unplanned shoppers are in the mall.

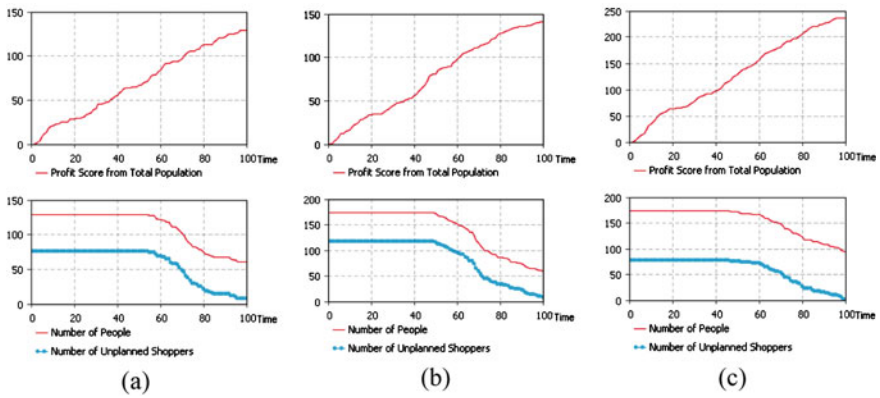


Fig. 4.10 Results for testing the impact of unplanned shoppers on the profit of the mall under a high density case [Pr(purchase) = 0.8]

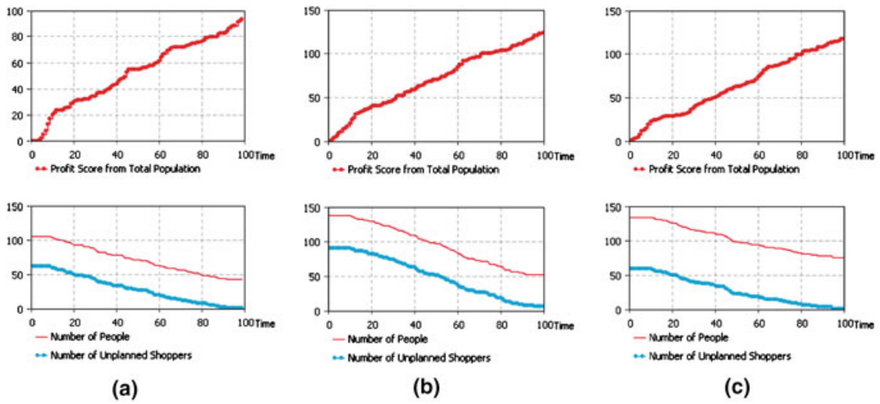


Fig. 4.11 Results for testing the impact of unplanned shoppers on the profit of the mall under high density case [Pr(purchase) = 0.6]

Figure 4.11c, however, demonstrates that 40 additional planned shoppers in the mall increases the profit score from 125 to nearly 250. When we decrease Pr(purchase) from 0.8 to 0.6, the profit does not increase much by adding either more unplanned or more planned shoppers. Besides, the profit from planned shoppers is almost the same as that from unplanned shoppers. In other words, the higher the buying intention, the more profit the mall will gain. Therefore, operational strategies such as promotion should target at increasing shoppers' buying intention.

Next, an experiment involving a low density environment has been also conducted with 29 planned shoppers and 39 planned shoppers (See Fig. 4.12a), where planned shoppers' purchase probability is 0.8. By adding 20 more unplanned shoppers (see Fig. 4.12b) and planned shoppers (see Fig. 4.12c) into the mall,

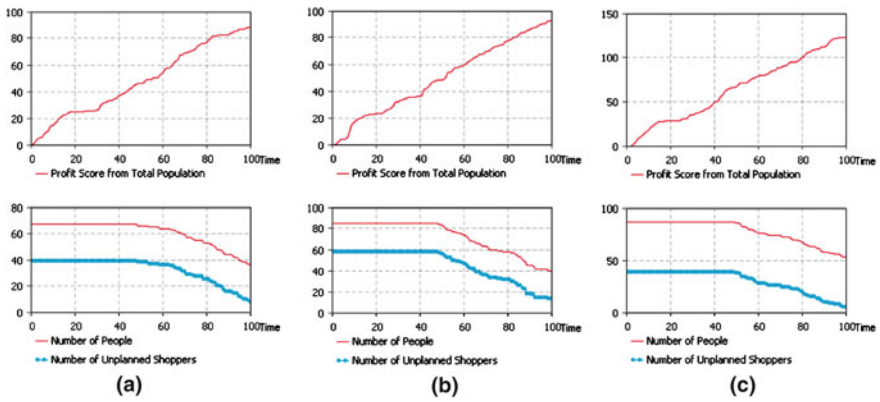


Fig. 4.12 Results for testing the impact of unplanned shoppers on the profit of the mall under a low density case [Pr(purchase) = 0.8]

respectively, smaller differences are observed compared with the case with higher density environment. According to our experiments, it has been found that the impact of the number of unplanned shoppers on the profit of the mall is more obvious when the mall is more crowded (e.g. during holidays or weekends). It is believed that this finding (and more detailed simulation results) would be very useful for the shopping mall management when they design promotion and/or advertisement policies during the regular as well as busy seasons.

4.4.3 Impact of Group Shopping Behavior on the Profit of Mall

As described in Sect. 4.3.4.2, individual shoppers would skip a planned shop or go to an alternative one (of the same type) if the crowdedness level in that shop is above their threshold. They make these decisions only based on their interests and how the shop attracts them (e.g. promotion). However, when a member in a group wants to skip a planned shop or go to an alternative shop, he/she needs to communicate (discuss) with all the other group members first and follow the group's final decision (which may accept or reject his/her proposal). Therefore, the chance that group shoppers skip or alter a shop is lower than that of individual shoppers. The goal of this experiment is to test our intuition that the shopping mall will gain more profit as the percentage of group shoppers increases. Figure 4.13a depicts the experimental results for the case with 48 individual planned shoppers, 50 group planned shoppers, and 89 unplanned shoppers. Here, the ratio between the group shoppers to the individual shoppers is about 1. Figure 4.13b depicts the experimental results for the case, where the number of unplanned shoppers remains unchanged, the number of group shoppers is increased to 70, and the number of individual shoppers is reduced to 34. It is clearly shown in Fig. 4.13b that the profit score increases as the percentage of the group shoppers increases.

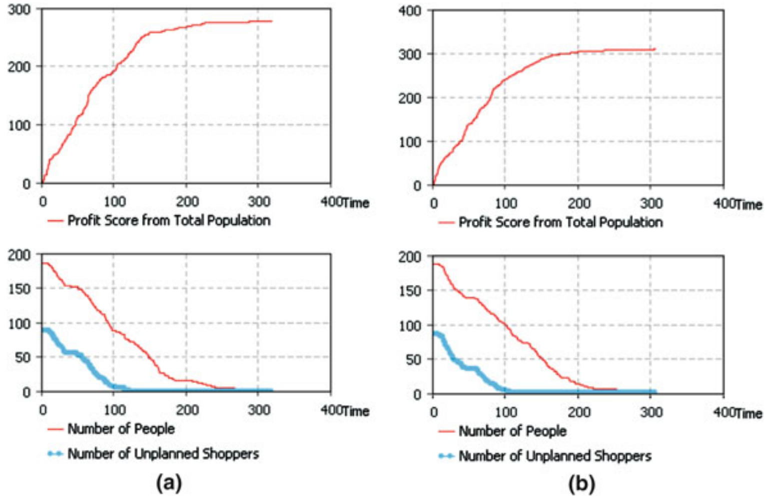


Fig. 4.13 Results for testing the impact of group shopping behavior on the profit of the mall

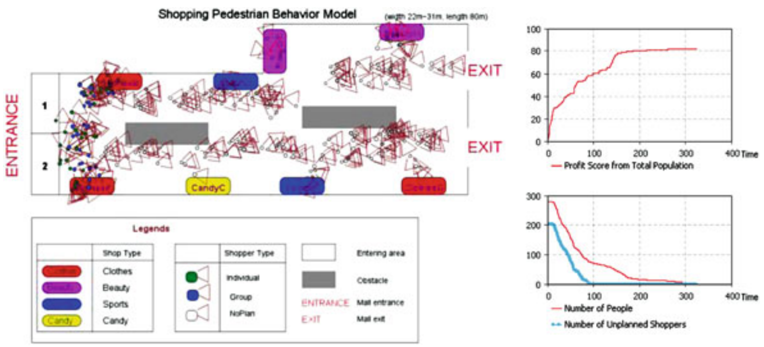


Fig. 4.14 Configuration 1: placement of similar stores far from each other

4.4.4 Arrangement of Stores

An empirical study conducted by Zhuang et al. (2006) demonstrated that the number of stores visited has a negative impact on shoppers' purchase. We observe that similar shops are usually located near to each other in many large shopping malls such as Dillards and Macy's. The goal of this experiment is to test the impact of the arrangement of stores in the considered shopping mall on their profit score. Two different configurations have been considered: (1) same-type shops are placed far from each other (see Fig. 4.14) and (2) same-type shops are placed close to each other (see Fig. 4.15). Experimental results reveal that the shopping mall gains a higher profit for the second configuration. One possible reason could be that

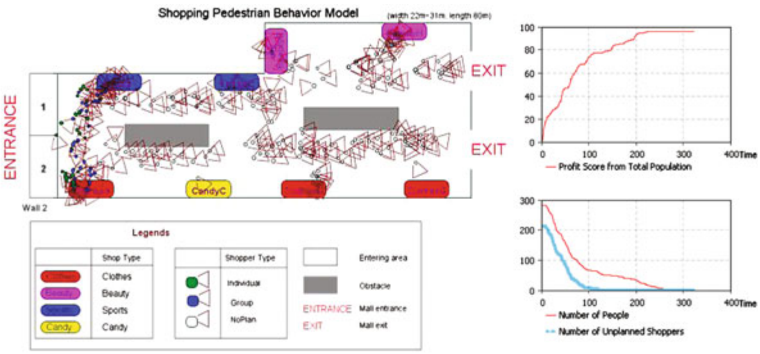
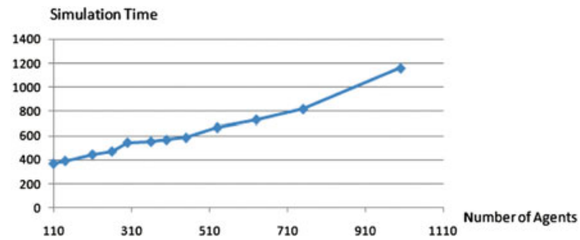


Fig. 4.15 Configuration 2: placement of similar stores together

Fig. 4.16 Simulation execution times with increase in the number of agents



people are more likely to make a purchase in a similar store nearby if their original planned destination is crowded. To validate this conjecture, further study such as survey on shoppers will be valuable.

4.4.5 Scalability and Computational Aspects

In this research, several efforts have been made to enhance the validity of the crowd simulation model for the considered shopping mall, such as (1) adopting EDFT to mimic decision deliberation of each individual pedestrian (at each point to choose a next direction), (2) incorporation of explicit group communications, and (3) consideration of human's vision into pedestrian's movement (SFM). However, it is expected that these additions will result in longer simulation execution times. The goal of this experiment is to test the scalability of the proposed, integrated pedestrian behavior model in terms of computational requirements. By increasing the number of agents involved in the simulation, we have evaluated the simulation execution times. As shown in Fig. 4.16, simulation execution times increase nearly linearly when the number of agents increases. Therefore, it is believed that our modeling approach is extensible to more complex situations without involving significant increase in the computational time.

4.5 Conclusion

The integrated pedestrian model proposed in this chapter has allowed us to develop a more realistic simulation of pedestrian behaviors at a shopping mall. In particular, consideration of the vision of each individual allowed us to mimic physical and psychological interactions among the people and the environment more realistically. Similarly, consideration of the EDFT (based on the psychological principle) allowed us to represent the human decision deliberation process, where economic approaches based on expected values are not always applicable. In addition, consideration of a rich set of attributes for the environment (different kind of shops, obstacles, promotions on the shops) as well as people (e.g. gender, age, preference, schedule, and grouping) has allowed us to mimic a real shopping mall environment closely. A crowd simulation model constructed based on the proposed pedestrian model and data (survey and observations) has been used to conduct several experiments. Our experimental results revealed several interesting findings such as (1) consideration of human vision into SFM (part of the contribution in this work) is found to be significant, (2) impact of shoppers' buying intention on the profit of the mall, especially when the mall is crowded (e.g. during holidays or weekends), (3) the profit score largely increases as the percentage of the group shoppers increases, and (4) the shopping mall gains a higher profit if similar-type shops are placed close to each other. It is believed that many of these findings (and more detailed simulation results) would be very useful for the shopping mall management when they make strategic decisions (e.g. layout design and arrangement of stores) as well as operational decisions (e.g. promotion and/or advertisement policies during the regular as well as busy seasons).

4.6 Future Work

Currently, the dynamic planning algorithm is based on a rather simple probability, but our future work will employ the extended decision field theory for shopping path planning according to the dynamically changing environment. In addition, while efforts have been made to collect data and validate part of the model via observations made at the mall, more comprehensive data collection and validation will be performed via human-in-the-loop experiment using the CAVE-based virtual reality environment. This environment will be used to simulate shopping mall situations under various conditions and collect information about the decisions made by individuals. The collected data will be used to support the development and calibration of the proposed pedestrian behavior model.

To this end, the first task will be to identify test scenarios covering a broad spectrum of different shopping environments to support model construction. Human-in-the-loop experiments will be executed using a scenario representing situations forcing shoppers to make a series of decisions. During shopping,

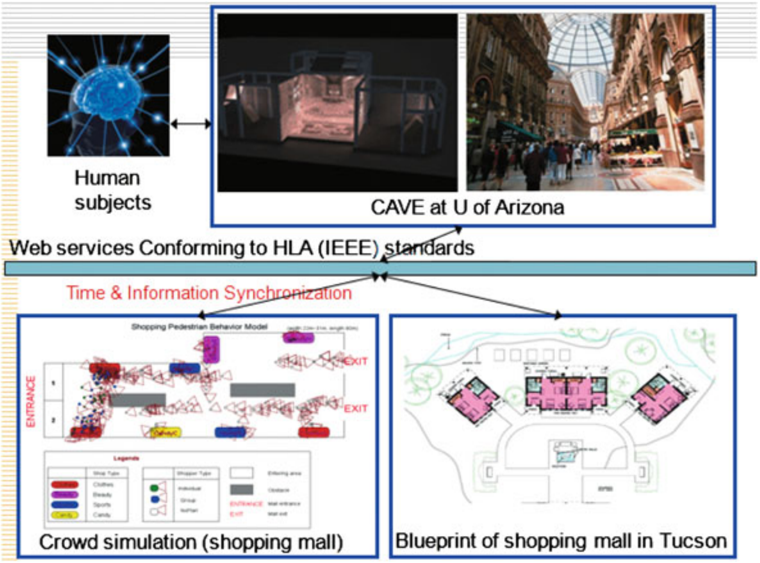


Fig. 4.17 VR environment (using integrated simulators) for human experiments

individuals must decide which stores to visit first according to the shopping plan (see Sects. 4.3.3 and 4.3.4). This decision often depends on various factors such as crowd density, tightness in their schedule, and real-time attractions (e.g. promotion and sales) from stores. Once a destination store is determined, shoppers must choose one of the six directions (see Fig. 4.3) at major decision points along the path. This decision often depends on the crowd density and the relative distance to the destination.

The second task will be to develop a computational model required to provide a realistic virtual test environment for the implementation of scenarios to conduct human-in-the-loop experiments. For the scenario, a high-fidelity shopping mall simulator will be set up to investigate how an individual shopper plans and makes decisions in a dynamic manner. During the experiments, shoppers will navigate in an area within the simulator based on their plan (see Fig. 4.8). At each decision point, each subject will be asked to evaluate the crowd density and performance (e.g. remaining distance to the destination) of available alternatives (e.g., choose a direction, choose a store to be visited next) depending on various observations. The effects of varying shopping mall conditions will be assessed by running experiments featuring, among others, different crowdedness, arrangement of stores, and various types of real-time information (e.g. sales and promotions) available. Experiments will be conducted using a CAVE three-dimensional virtual reality environment. In such an environment, subjects sense that either the user's point of view or some part of the user's body is contained within the computer-generated space. This allows observing quasi-real human responses. The hardware system that will be used is the FakeSpace simulator, located at the University of

Arizona, and which has already been successfully used by the authors to assess evacuation behaviors under a terrorist bomb attack (Lee and Son 2008; Shendarkar et al. 2008), evacuation behaviors under fire in a factory (Vasudevan and Son 2008), virtual stock investment (Son and Jin 2006), and error detection and resolution by people in a complex manufacturing facility (Zhao and Son 2008).

The third task will be to employ efficient and effective synchronization and coordination mechanisms (which the authors have already developed) for linking simulation elements that will be hosted at different computers. To enhance the realism of the shopping mall simulator executed in the CAVE environment, it will be federated in real-time with other simulators (e.g. crowd simulator) via the web services (see Fig. 4.17).

References

Babin B, William D et al (1994) Work and/or fun: measuring hedonic and utilitarian shopping value. J Consumer Res 20(4):644–656

Batra R, Ahtola O (1991) Measuring the hedonic and utilitarian sources of consumer attitudes. Mark Lett 2(2):159–170

Baumann D, Robert C et al (1981) Altruism as hedonism: helping and self-gratification as equivalent responses. J Pers Soc Psychol 40(6):1039–1046

Blue VJ, Adler LJ (2001) Cellular automata microsimulation for modeling bi-directional pedestrian walkways. Transp Res Part B 35:293–312

Busemeyer JR, Diederich A (2002) Survey of decision field theory. Math Soc Sci 43:345–370

Busemeyer JR, Townsend JT (1993) Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. Psychol Rev 100:432–459

Gaskell GD, Benewick RJ (1987) The crowd in contemporary Britain. Sage, London

Hamagami T, Hirata H (2003) Method of crowd simulation by using multiagent on cellular automata. In: Proceedings of IEEE/WIC international conference on intelligent agent technology (IAT'03), Halifax, Canada, pp 46–52

Helbing D, Farkas I et al (2000) Simulating dynamical features of escape panic. Nature 407: 487–490

Helbing D, Buzna A et al (2005) Self-organized pedestrian crowd dynamics: experiments, simulations, and design solutions. Transp Sci 39:1–24

http://www.w3c.org

Hu Q, Fang W et al (2009) The simulation and analysis of pedestrian crowd and behavior. Sci China Ser E Technol Sci 52(6):1762–1767

Kuruvilla S, Joshi N et al (2009) Do men and women really shop differently? an exploration of gender differences in mall shopping in India. Int J Cons Stud 33:715–723

Lee S, Son Y (2008) Integrated human decision behavior modeling using extended decision field theory and soar under BDI framework. IERC, Vancouver, Canada

Lee S, Son Y et al (2005) Decision field theory extensions for behavior modeling in dynamic environment using Bayesian belief network. Inf Sci 178:2297–2314

Lee S, Zhao X et al (2008) Fully dynamic epoch (FDE) time synchronization method for distributed supply chain simulation. Int J Comput Appl Technol 31(3–4):249–262

Ma J, Song W et al (2010) k-Nearest-neighbor interaction induced self-organized pedestrian counter flow. Phys A 389:2101–2117

Moussaïd M, Helbing D et al (2009) Experimental study of the behavioural mechanisms underlying self-organization in human crowds. In: Proceedings of Royal Society B, vol 276, pp 2755–27

Muramatsu M, Irie T et al (1999) Jamming transition in pedestrian counter flow. Phys A 267:487–498

Parisi D, Gilman M et al (2009) A modification of the fial force model can reproduce experimental data of pedestrian flows in normal conditions. Phys A 388:3600–3608

Rathore A, Balaraman B et al (2005) Development and benchmarking of an epoch time synchronization method for distributed simulation. J Manuf Syst 24(2):69–78

Shendarkar A, Vasudevan S et al (2008) Crowd simulation for emergency response using BDI agents based on immersive virtual reality. Simul Model Pract Theory 16:1415–1429

Son Y, Jin J (2006) Extended BDI framework and technologies for modeling partial human decision-making. AFOSR Cognition & decision program review workshop, Fairborn, OH

Vasudevan K, Son Y (2008) Concurrent consideration of evacuation safety and productivity in manufacturing facility planning using multi-paradigm simulations. 18th international conference on flexible automation and intelligent manufacturing, Skovde, Sweden

Xia Y, Wong SC et al (2009) Dynamic continuum pedestrian flow model with memory effect. Phys Rev 79(066113):1–8

Zacharias J, Bernhardt T et al (2005) Computer-simulated pedestrian behavior in shopping environment. J Urb Plan Dev 131(3):195–200

Zhao X, Son Y (2008) BDI-based human decision-making model in automated manufacturing systems. Int J Model Simul 28(2):1–10

Zhuang G, Tsang A et al (2006) Impacts of situational factors on buying decisions in shopping malls. Eur J Mark 40(1/2):17–43

Chapter 5
Determining the Number of Simulation Runs: Treating Simulations as Theories by Not Sampling Their Behavior

Frank E. Ritter, Michael J. Schoelles, Karen S. Quigley and Laura Cousino Klein

Abstract How many times should a simulation be run to generate valid predictions? With a deterministic simulation, the answer simply is just once. With a stochastic simulation, the answer is more complex. Different researchers have proposed and used different heuristics. A review of the models presented at a conference on cognitive modeling illustrates the range of solutions and problems in this area. We present the argument that because the simulation is a theory, not data, it should not so much be sampled but run enough times to provide stable predictions of performance and the variance of performance. This applies to both pure simulations as well as human-in-the-loop simulations. We demonstrate the importance of running the simulation until it has stable performance as defined by the effect size of interest. When runs are expensive we suggest a minimum number

F. E. Ritter (✉)
College of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802, USA
e-mail: frank.ritter@psu.edu

K. S. Quigley
Department of Psychology, Northeastern University,
125 Nightingale Hall, Boston, MA 02115, USA

Department of Veterans Affairs, Edith Nourse Rogers (Bedford)
Memorial VA Hospital, Bedford, MA 01730, USA

L. C. Klein
Department of Biobehavioral Health,
The Pennsylvania State University, PA, USA

M. J. Schoelles
Department of Cognitive Science, Rensselaer Polytechnic Institute, Troy, NY, USA

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_5, © Springer-Verlag London Limited 2011

of runs based on power calculations; when runs are inexpensive we suggest a maximum necessary number of runs. We also suggest how to adjust the number of runs for different effect sizes of interest.

5.1 Introduction

We provide guidance here on how many times to run a simulation, including a human-in-the-loop simulation or a cognitive model, to ensure that the simulation has converged on stable predictions. This advice is derived from power calculations.

One paradigm in which this heuristic guidance can be applied is when a model is constructed to perform and make predictions about some human task. Although this seems like a narrow topic to explain to a general simulation audience, the number of times to run a simulation is an important topic because in many cases simulations are being used incorrectly, and as a result, analysts and their audience do not truly understand the simulation's predictions. We begin with our view of simulation. This will require talking about several layers of a simulation taxonomy until we reach the level at which this chapter is aimed, and then illustrating the problem and quantifying the solution for an example simulation.

The methodology we prescribe provides suggestions for any simulation with random processes as components, including the development of human-in-the-loop simulations and stochastic cognitive models developed and run on computational cognitive architectures. In the subsequent sections, we will elaborate on these terms.

The first term to note in our taxonomy is "computational". What is the difference between computational modeling and statistical or mathematical modeling? We believe that the main difference is that computational models are computer programs rather than equations or distributions. The advantage of models as computer programs is that they can simulate complex behavior. For instance, a computational model of a human playing a computer-generated game of Tetris through a computer interface is feasible, but it seems it would be very difficult to develop a mathematical model of the integrated cognitive, perceptual, and motor processes involved in this task. One important advantage of models as computer programs is that they can be process models, providing a theory of the information processes in cognition by processing information themselves.

Computer programs intended to model some cognitive process belong to a part of artificial intelligence (AI) called human-level AI or cognitive science, depending on the emphasis of intelligence level or being human-like in the processing. One approach being taken to achieve human-level intelligence is simulation of human behavior. The field of cognitive architectures has developed in the last 25 years to create high-fidelity simulations of human behavior. There are many definitions of the term cognitive architecture. Most definitions include some notion that the cognitive architecture contains the immutable functional machinery of cognition. For example, a definition by Ritter and Young (2001), consistent with Newell (1990) is:

> A cognitive architecture embodies a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task.

That is, those information processing computations are not modified by changes in beliefs, goals, and so on.

Another important notion of cognitive architectures is that the architecture by itself cannot produce any behavior. Knowledge must be added to the architecture to achieve behavior, creating a model. The current state of the art in cognitive architectures is that the modeler must supply the knowledge. Therefore, many architectures equip the modeler with a modeling language to develop models. While being able to program models via a modeling language has benefits in terms of efficiency and complexity, Byrne (2003) points out that "individual modelers need to have solid programming skills". The advantage of architectures implemented as computer programs is that the programming language is not ambiguous, and therefore supports a more uniform interpretation of the theory.

Cognitive architectures also come in many flavors. Cognitive architectures as computer programs represent scientific theories such as those in ACT-R, Soar, and EPIC. Some like Soar (Laird et al. 1987) and EPIC (Kieras 2003; Kieras et al. 1997) are basically deterministic in the sense that in most models noise is not directly added to computations and the same predictions are made each time the model is run. ACT-R is an example of a hybrid architecture that has stochastic components. The example model described later in this paper is an ACT-R model (Anderson 2007).

To understand the dilemma presented in the following section a brief description of the ACT-R architecture is required. ACT-R is symbolic and rule based, but also has a layer below the symbolic layer, called the sub-symbolic layer. One quantity computed by the subsymbolic layer is the activation of declarative memory elements. This activation determines the retrieval probability and latency for a memory element. The other important computed quantity is the rule utility based on temporal difference reinforcement learning. The computation of both these quantities involves the addition of noise to the calculation. These noise quantities are controlled by the modeler through parameters (i.e., one for activation and one for utility). Therefore, ACT-R models can be stochastic and most are, unless major components are removed by setting the noise to zero.

To develop a model in ACT-R, the model adds procedural knowledge in the form of production rules and background knowledge in the form of declarative memory chunks. This allows very complex models to be built, but these types of models can be difficult to analyze and evaluate because of the inherent complexity of the knowledge contained within them and the variability in processing the knowledge by the noisy architecture.

As discussed above, cognitive architecture-based models are often built to simulate human users of computer systems. So, to evaluate such a model it seems natural to want to compare model data to human data. The traditional way to do this is by hypothesis testing where the null hypothesis is that there is no difference between the human data and model data. But hypothesis testing can only be used to reject the null hypothesis. Thus, we can show that model data does not match the human data, but cannot prove that it matches. (Grant 1962 provides an argument showing how correlation helps to provide an answer in this area.)

The problem of how many times to run a model is one part of the bigger problem of model comparisons. One hoped for outcome of this handbook is to provide guidance to modelers developing complex models on how to show that the model data corresponds to human data. In a symposium on “Model Fitting and Parameter Estimation” Ritter (2003) posited the following points on the problem of model validation and comparison for the types of models developed under cognitive architectures: (a) Task performance is more important than fit—more credit should be given to models that actually perform the task. (b) Enough detail should be given about the model fit to see whether the model is worth taking seriously or not. If the model can fit any data, then it should be dismissed as a psychology theory (but may be useful as an AI model). (c) It should be reported where the model can be improved. In other words, let the reader know where the holes are and where the model can and will be improved. This view is that of Grant (1962) as applied to cognitive architecture-based models. But again, before one can think about model comparisons, the model’s predictions must be understood.

One of the strengths of ACT-R and architectures like it is the ability to interact with the same software as humans-in-the-loop. It can do this because it has “eyes, hands, ears”, and can speak. These perceptual and motor components of ACT-R are not only psychologically plausible but can interact with the stimulation and operating system software to manipulate input devices and read the computer screen. With these components ACT-R models perform human actions such as searching computer screens, listening to instructions, and manipulating a mouse or joystick.

The relevance to human-in-the-loop systems is that ACT-R can be a human-in-the-loop when human subjects are expensive. Imagine a team-oriented simulated task environment, where the team members are at workstations and communicate over a network. ACT-R models can be developed to work in such environments, replacing one or more of the team members, or, for some studies, all of the team (e.g., Ball et al. 2010). The issue for this paper is how many times do you need to run such a simulation to understand the implications of the simulation, with or without humans in the loop?

5.2 Modeler’s Dilemma

Because cognitive models are really simulations, a common question facing creators of cognitive models, at least implicitly, is “How many times should we run the model to generate predictions for comparison across experimental conditions and for comparison with the data?” As we note below, authors have used a wide variety of answers: some comparisons use a single run of the model, although this is somewhat uncommon with models that include stochastic effects. Some comparisons run the model once per subject. This is often just a handy heuristic as they

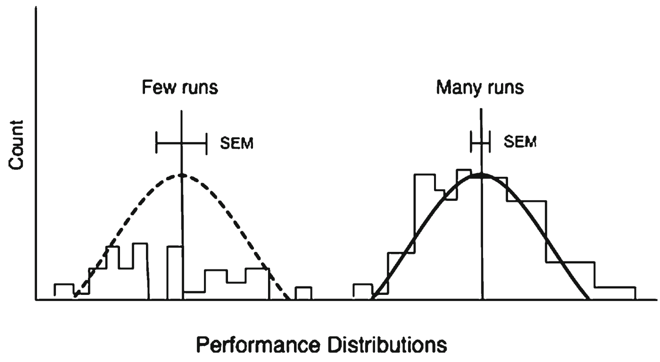


Fig. 5.1 The distribution of performance, mean, and standard error of the mean for a model run a few times (*left*) and run many times (*right*). The distribution for the few runs is *dashed* to show that it is a less accurate representation

look for a number to choose. Other researchers run it 10, or 20, or 50, or 1,000 times. The dilemma is that you want to run a stochastic model enough times to understand its predictions without running it so many times as to waste resources. In completely human studies, this problem is addressed through power calculations. For other simulations including human-in-the-loop simulations, power calculations would be useful as well.

Figure 5.1 illustrates this problem. On the left-hand side, if a model with random elements is run only a few times, the distribution of performance is not well known (shown with a shaded line indicating a less understood distribution). The mean and standard deviation are also less well known, and the standard error of the mean,[1] is larger. On the right-hand side of Fig. 5.1, where the model is run more times, the distribution is better known (shown with a more complete histogram and a solid estimated distribution line). The mean and standard deviation become more stable and the standard error of the mean becomes smaller with additional runs. And yet, with further runs the improvement that each run provides decreases.

The extent of the problem of knowing how many times to run a model can be illustrated by looking at a sample of models. There are many venues where this can be done. Table 5.1 provides just one example set, a summary of models presented at the 2004 International Conference on Cognitive Modeling (Lovett et al. 2004) where the papers are available online. Similar results are available for other sets of models.[2] The table includes each paper reporting a model to data comparison

[1] The standard error of the mean is a standard statistical measure of how well known the mean is, and it is explained in more detail below.

[2] For example, http://acs.ist.psu.edu/nottingham/eccm98/home.html

Table 5.1 Number of model runs compared to subject data for papers at the 2004 International Conference on Cognitive Modeling (Lovett et al. 2004)

Paper	Subjects	Model runs
Altmann and Burns (2004)	71	ng
Belavkin and Ritter (2004)	ng	ng
Brumby and Howes (2004)	20	100
Byrne et al. (2004)	164	100
Chandrasekharan et al. (2004)	3	10
Chartier et al. (2004)	ng	100
Chavez and Kimbrough	48	20
Chong (2004)	ng	ng
Cox and Young (2004)	ng	ng
DelMisser (2004)	60	~8
Fu et al. (2004)	32	ng
Fum and Stocco (2004)	ng	ng
Gray et al. (2004)	54	48
Halverson and Hornof (2004)	24	2,520
Kushleyeva et al. (2004)	10	10
Maka et al. (2004)	45 essays	ng
Marnier and Laird (2004)	na	100
Martin et al. (2004)	11	20
Matessa (2004)	ng	ng
Matusuka and Corter (2004)	14, ng	50, 500
Morita and Miwa (2004)	33	ng
Nason and Laird (2004)	na	500
Nellen and Lovett (2004)	160	180
Nuxoll et al. (2004)	na, na	5, ng
Nuxoll and Laird (2004)	na	5
Peebles and Bothell	30	150
Rutledge and West (2004)	3	1,000
Salvucci et al. (2004)	11	ng
Simen et al. (2004)	3	ng
St. Amant and Ritter (2004)	6	20
Stewert et al. (2004)	2,571	1,000
Taatgen et al. (2004)	ng	ng
Wu and Liu (2004)	ng	7,200

ng not given. *na* not applicable, as model results were presented for illustration only or the model was not stochastic

where the model appeared to have a stochastic component or where the task provided variance. The second and third columns note how many subjects were included and how many times the models were run.[3]

[3] Papers with two studies had each study counted 0.5. Papers that were not simple, that examined complex data, e.g., language corpora, or that presented only tools or theoretical points, are not included

Table 5.1, which is representative of other conferences and even journal papers, shows that more than a third (12.5/33) of the papers did not report on how many times the model was run; and an additional 7.5 probably did not run their model enough to report stable predictions (20 or fewer runs). So, well over half did not run their model to get stable predictions or did not report that they did. In addition, none of the papers in Table 5.1 provided a rationale for the number of model runs beyond "to match the number of subjects" or "to provide stable performance". No paper mentioned effect sizes, although many included standard error bars on their graphs. The number of times the models should have been run is not known to us—it would depend on the effect size of interest, but we will see that it is most likely that the number of runs was too low. (The number of runs needed would also vary based on the number of parameters manipulated, but these models did not vary parameters or perform parameter sweeps). This lack of reporting of the theories is alarming.

Of course, where models are deterministic, they only need to be run once. Where there are closed-form solutions to obtain the predictions of models, these closed-form solutions should be used. For example, we have run a Soar model for 100 h to compute predictions, only to discover with a bit of mental effort that a closed-form iterative function would provide the same data in 6 s on a slower machine (Ritter 1988).

When runs are inexpensive, using a very large number of runs (e.g., 10,000–100,000) is a very satisfactory answer because it provides stable estimates of performance, and the power analyses below indicate why. However, there are an increasing number of cases when simply performing a large number of runs will not work. Performing a large number of runs is not possible when runs are expensive, numerous models must be run as in a network, or search in a combinatorial parameter space is required (where there may be 100,000 parameter sets to test, making 1,000 runs per setting turn into 100,000,000 runs). These situations include models that interact with or are based on robots that are both complicated to set up and cannot be run faster than real time, models that work with commercial software that can only run in real time, models that interact over a long time period, models that have multiple settings or parameters to be adjusted, models that interact with software too complicated to rewrite to run faster than real time (e.g., some process control models), and models that have to interact with people (i.e., human-in-the-loop simulations) or simulate group behavior with real time constraints (e.g., they cannot be run faster than real time).

Even when models can be run faster than real time there are cases when the modeler might wish to run as few as necessary. These include when there are multiple models to be considered or a combinatorial set of possible parameter sets (e.g., changes to working memory, changes to processing speed, and changes to representation). Even for models running faster than real time one should ask how many runs are needed to understand the model's predictions?

We will present the case here, using an example representative model that suggests that researchers should run their model until it makes stable predictions (that is, the predictions obtained are representative of the model's predictions). We will also describe a way to compute stability. Our results suggest that some of the models in Table 5.1 may have been appropriately presented, but most could have been understood better and presented more clearly by following the suggestions we make below. We provide a rationale and a way to compute how many runs represent stable predictions based on effect sizes and the power with which the modeler wants to determine these effects. Our approach also shows that an answer of providing "an infinite number of runs or as many as possible" (which could also be put forward) is a wasteful and unnecessary prescription for human-in-the-loop simulations. For illustration we use a medium-sized model we created to understand behavioral moderators. We analyze its behavior as an example—the calculations and implications apply to all user models with stochastic elements. We introduce that model next.

5.3 Example Model: Cognitive Appraisal and Subtraction

Serial subtraction commonly has been used to assess the relationship between task appraisals and resulting physiological changes. This task is regarded as an active coping task and has been used across many laboratories as a stressor task (e.g., Kirschbaum et al. 1993; Quigley et al. 2002; Tomaka et al. 1993). In this task, subjects are given an arbitrary seed number and are asked to subtract repeatedly a single- or double-digit number. For example, a subject is given 1,457 as the seed number and is asked to repeatedly subtract seven from the running total while speaking aloud each result. Mistakes are noted and the subject is asked to correct them before they can continue.

The type of appraisal made prior to the task affects the performance on the serial subtraction task—subjects making challenge appraisals attempt more subtractions and have more correct responses than subjects who make threat appraisals prior to the task (Tomaka et al. 1993). A "challenge" appraisal occurs when, although stressfulness of the task is deemed high, coping ability is also deemed high. A "threat" appraisal occurs when stressfulness is high and coping ability is seen as low. Although serial subtraction may not appear stressful to everyone, it is typically challenging and often threatening to participants, probably due in large part to the highly evaluative and social nature of the task (e.g., the experimenter often is seated close to the subject and "knows the answers", and the subject is told that they are being recorded for "later analyses"). We know that these appraisals influence the performance and are not entirely evaluations of knowledge because performance varies when the participant's appraisals are manipulated and knowledge is held constant (Tomaka et al. 1997).

5.3.1 *The Model*

To illustrate the effect of increasing the number of model runs we chose a cognitive model of serial subtraction that was built using the ACT-R cognitive architecture. It is similar in size and complexity to many ACT-R models and models being developed in other architectures. ACT-R is a production rule-based cognitive architecture; that is, cognitive activity takes place through the successive firing of production rules that take an “if…then” format. The model includes several stochastic elements. The details are not important for this analysis, but are available in the descriptions of the model (Ritter et al. 2004, 2007b) and of the architecture (Anderson and Lebiere 1998).

The choice of which rule to fire from among those that match a particular situation (so-called conflict resolution) is thus a knowledge-based process, where higher valued rules represent more strongly held beliefs. It is also a stochastic process due to the presence of ACT-R’s expected gain noise (EGN) parameter. This noise allows the occasional firing of rules that are less than optimal. Adding noise to the decision process is consistent with several theories of stress indicating that high levels of stress negatively influence cognition, particularly decision making (e.g., Mathews 2001), and as we shall show, consistent with existing data. There are, of course, other possible approaches to modeling stress in ACT-R (Ritter et al. 2007b). Testing all of them and all of their combinations would be a useful but non-trivial exercise, so this model is presented here for illustration. Indeed, the need to test these combinations (up to 200 possible variants) suggests the need to understand how many times we need to run each variant to understand its predictions.

Our current serial subtraction model contains the necessary procedural knowledge (i.e., 28 rules implementing subtraction) to perform the serial subtraction task as well as declarative knowledge about numbers and arithmetic facts (257 declarative memory elements made from 16 types, such as digits, columns, subtraction-facts, and comparison of number pairs). The model, the graphical interface, and movie demos of the model running are available (acs.ist.psu.edu/ACT-R_AC/).

5.3.2 Changes to the Model Examined

The capacity for the model to perform the task under threat or challenge appraisals is implemented by adjusting the value of the rule utility noise parameter[4] to simulate the effects of cognitive appraisal influencing the decision process about what knowledge to apply. When the model is set to challenge appraisal, the rule utility noise parameter is set to a small value (0.1) to model a “clear head”, but where errors can occur as they do in real subjects. When the model is set to threat appraisal, the default value of the rule utility noise is changed to a greater number (1.0) to simulate a state where the procedural knowledge is applied less accurately in the thought process of threatened individuals. Although appraisals are often dichotomized as challenge or threat, they fall along a continuum of appraisals and

[4] The parameter is EGN in ACT-R 5, and EGS in ACT-R 6.

thus this parameter could vary across a continuum as well. An attractive feature of this type of modification based on modifying architectural parameters is that it is based on a cognitive architecture (Ritter et al. 2007b). This allows the modification to be easily borrowed and used by any other model built in ACT-R. We have applied a related change to a model of driving (Ritter et al. 2006).

These changes to produce two conditions of the model, however, are used for illustrative purposes here. Our analysis applies to any model that makes predictions that include a stochastic component, and where closed form or infinite runs are not available.

5.4 Comparison of the Model with Data

Results from the model performing the serial subtraction task under challenge and threat appraisals can be compared to human data obtained from an empirical study using the same task. The first three rows of Table 5.2 present the human data taken from Tomaka et al. (1993) of subjects performing the serial subtraction task who made pre-task appraisals. With more challenging appraisals, more subtractions were attempted and more attempts were correct. (These differences were reported by Tomaka et al. as being significantly different, but standard deviations were not reported in their paper. The ACT-R model predicts that the standard deviations were small with respect to their sample size and mean and that these differences are reliable.) The model's standard deviations are, however, much smaller than data from later studies where the SD (across subjects) is approximately 15 subtraction attempts (Ritter et al. 2006).

The model's predictions with the pre-task appraisal overlay are shown in the second set of rows of Table 5.2 (rows 4–6). In each case, the model makes predictions that are different from each other ($p < 0.01$) for each type of appraisal. The model for threat appraisals reproduces fairly accurately the average number of attempts and correct responses when performing under threat appraisal. However, in the case of a challenge pre–task appraisal, the model does not perform as many subtractions as in the human data, but successfully matches the ratio of correct responses to subtraction attempts.

The model's performance was measured over multiple runs because its performance varied. The noise applied to the model is supplied by a pseudorandom number generator based on the Mersenne Twister, which is designed to provide numbers with low autocorrelation, that is, runs of ACT-R are independent when taken in a series, and are samples from a single distribution (independent and identically distributed). When the rule utility noise is zero (EGS = 0), the model exhibits perfect performance because it applies rules completely accurately. When the rule utility noise is greater than zero, the model can make several kinds of mistakes based on applying a nearly appropriate but wrong rule, or applying the right rule at the wrong time. The rules chosen can vary slightly (at EGS = 0.1) to somewhat (EGS = 1.0) from optimal.

Table 5.2 Comparison of the model's behavior for threat and challenge conditions to human data taken from Tomaka et al. (1993) per 4-min block

Cognitive Appraisal Conditions				
		Threat	Challenge	
Human data ($N = 22$)	Attempts	46	61	
	Correct	42	56	
	% correct	91	92	
		Threat (EGS = **1**)	Challenge (EGS = **0.1**)	ACT-R Default (EGS = **0**)
Model (***N* = 100**)	Attempts	46.8 (3.6)	< 54.5 (3.5)	< 70.9 (1.3)
	Correct	42.5 (5.1)	< 50.2 (5.1)	< 70.9 (1.3)
	% correct	91	92	100

Standard deviations of the model's performance are shown in parentheses. '<' denotes significant difference at $\alpha = 0.01$

The good fit of the model with the pre-task appraisal overlay to the human data suggests that our choice of how to implement cognitive appraisal was sensible. The model offers one plausible and very simple hypothesis to explain the impact of cognitive appraisal on task performance. It encourages more work to determine whether the way appraisal affects performance is indeed by influencing the level of noise present in the thought process of humans.

But is this a fair and sufficient comparison of the model's performance with the data? How many times should we have run our model to confidently report its predictions? When we have multiple possible changes to our theories, how many runs do we need to test each of these modifications?

5.5 Computing How Many Runs to Perform

Figure 5.2 starts to answer the question of how many times a model should be run. It shows the individual number of subtraction attempts across 100 runs (light points) as well as the running, cumulative average values (dark points). The error bars are the cumulative standard deviation at each point, that is, the standard deviation for the points up to that run. Figure 5.2 illustrates the range of possible values, the problem of using just a few runs, and how with an increasing number of runs the true average is approached.

We propose two possibilities for a criterion for stable predictions. The first is based on the standard error of the mean. Figure 5.3 shows how the standard error of the mean (SEM) decreases over a series of 100 runs for our model.

Equation 5.1 shows how the SEM is based on the variance and size of the sample.

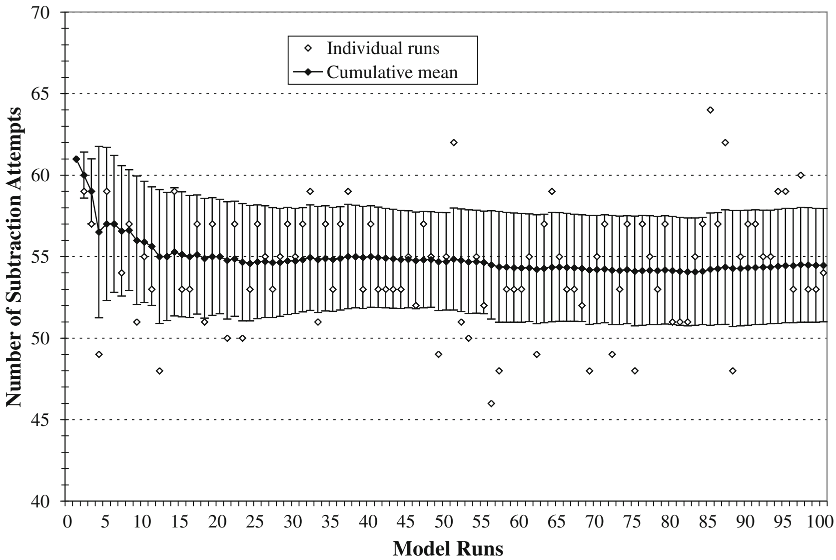


Fig. 5.2 The predicted number of total attempts and cumulative standard deviation as error bars across the 100 runs of the model with a challenge setting

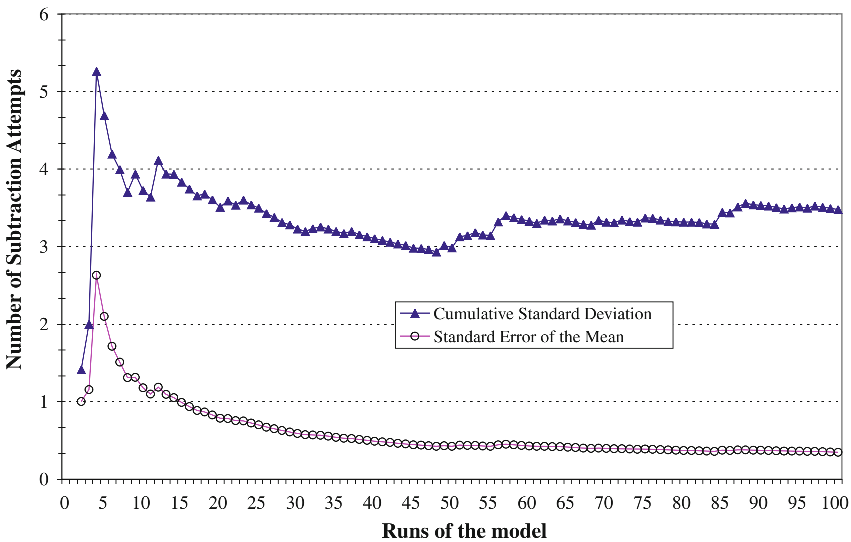


Fig. 5.3 The cumulative standard deviation and the cumulative standard error of the mean across 100 runs of the model with a challenge setting

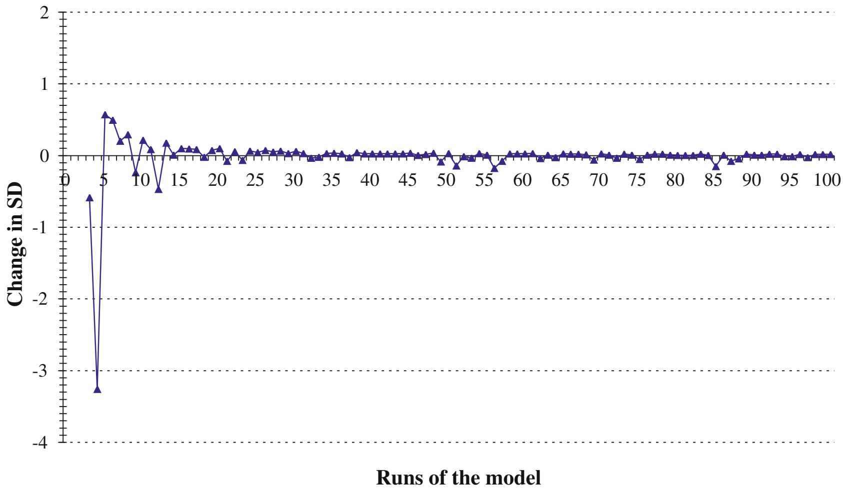


Fig. 5.4 The change in standard deviation (between run N and $N-1$) across the 100 runs of the model with a Challenge setting

$$\text{SEM} = \text{Variance} \,/\, N \;=\; \text{Standard deviation} \,/\, \text{sqrt}\,(N) \tag{5.1}$$

The SEM represents the error in predicting the mean of a distribution. Assuming that the values are independent and identically distributed, the SEM indicates that the true mean has a 95% chance of being within range of the estimated mean ±1.96*SEM. Thus, one way to determine how many times to run a simulation is to run it until the estimated range of the mean is small enough for your purposes.

Figure 5.3 also shows how the standard deviation stabilizes with additional runs. This figure is interesting because it also shows how the predictions of the mean and variance become more accurate with additional runs. The mean (Fig. 5.2) and the variance (Fig. 5.3) are initially unstable with a small sample of runs. With additional runs, the SEM basically decreases from run four on. With 100 runs the SEM is at 0.35 and decreasing rather slowly (related to the square root of the number of model runs, per Eq. 5.1). Figure 5.4 shows how the change in SD between runs decreases across the 100 runs.

In this case, if we wanted to know how many subtraction attempts the model predicted for a 4-min block, ± 0.5 subtractions with 95% confidence, based on Eq. 5.1 we would have to have a SEM of 0.5/1.96 or a SEM of 0.255 (0.5 = 1.96*SEM, or 0.5/1.96 = SEM = 0.255). If we use an estimate from Fig. 5.3 of the standard deviation as being 3.6 (it is probably slightly less), then 3.60/sqrt (N) = 0.255. Solving for N gives us 199 runs.

Together, Figs. 5.2 and 5.3 demonstrate that reporting a single run of our model, in particular the first run in our series, 61 attempts in a 4-min block, would have over predicted the number of attempts by about 10%. Other single runs would be more or less accurate. Some papers have reported one run of a model as an example. With deterministic models, this is appropriate. For models with stochastic components, these figures show that one run is clearly not enough.

Other reports have run the model once per subject. For this data set, the model would be run 22 times. Figure 5.3 suggests that the first 22 runs in our series would provide a fairly reasonable prediction of the mean total attempts from the model, 54.86. This prediction is still slightly high, however. Figure 5.3 goes on to show that with more runs, the model's average number of attempts drops slightly to 54.4 attempts. Figure 5.2 also shows that the SEM at 22 runs is 3.54/sqrt (22) = 0.75, and thus other sets of 22 runs could more or less accurately represent the model's performance.

The heuristic of one run per subject ignores that model runs are typically much less expensive than subject time. Moreover, different sets of 22 runs could lead the modeler to a wide range of different conclusions, which is clearly not desirable. Most importantly, if one takes the model to be a theory, then the choice of "number of runs = number of subjects" reports a sample of the theory rather than the theory's predictions and thus is not at all appropriate.

Figures 5.2 and 5.3 show that increasing the number of runs improves the quality of the model's predictions in that they are more accurate. Namely, the cumulative averages are more stable, the mean standard error decreases, the standard deviation stabilizes, and the corresponding power to find differences between model conditions and between the model's predictions and the data increases. The two figures suggest that the best number of runs for the model is simply the largest number possible, as more runs provide more stable and accurate predictions, although there are decreasing returns with more runs.

If one is using a simulation where running the simulation is not free or even inexpensive, one will have to choose a cutoff, however. For instance, using the model here, SEM drops to 0.5 by about 40 runs and then drops slowly with additional runs. So, Figs. 5.2 and 5.3 might lead to a different conclusion, which is to do runs until the changes to the mean and SD from run to run become negligible—where negligible is defined by the modeler and the size of differences of interest. For human-in-the-loop simulations, this might be 10, or 40, or more but you can see the trade-offs in the figures.

But, when simulation runs are not easy to obtain, how can we choose an appropriate number of runs? Power calculations are a way to compute how likely a study is to find effects based on their size and the size of the sample (Cohen 1988). This is the same computation that experiment designers use to determine how many subjects to run. It is a simple equation that can be used to compute the probability of finding a given effect size given the number of times a variable is measured. An effect size is the ability to see a difference between two means using the standard deviation as a unit. Thus, an effect size of 1 is observed when the difference between two means is separated by 1 standard deviation; 0.5 is a difference of half an SD, and so on. Because effect sizes are represented in terms of standard deviations, it does not matter what the source of noise is in the model,

Table 5.3 Power of t-tests ($\alpha = 0.05$, two-tailed) for a range of effect sizes

Mean Effect Size	N	δ	Power
0.1	100	0.71	< 0.17
0.2 (Cohen's small)	100	1.41	0.29
0.5 (Cohen's medium)	100	3.54	0.94
0.8 (Cohen's large)	100	5.66	> 0.99
2.14 (effect size reported here in the subtraction model)	100	15.13	> 0.99

This table uses δ = effect size * sqrt (N/2) (Howell, 1987, pp. 201–202, and associated appendix to compute power for the value of δ).

or whether the standard deviation is large in relation to the mean. A disadvantage to using effect sizes is that they are in terms of standard deviations, and not in the raw measure. For example, an effect size on reaction time is in standard deviations rather than milliseconds, which is slightly harder to reason about.

The equation for computing power is used in Table 5.3 and shown here as Eq. 5.2.

$$\delta = \text{effect size} * \text{sqrt}\,(N/2) \tag{5.2}$$

In this equation the noncentrality parameter (δ) is based on two components, effect size and sample size. For a given power value, small effect sizes require correspondingly larger sample sizes.

The SD of model performance (e.g., shown in Figs. 5.2 and 5.3) and sample size can be used to compute a measure of (statistical) power (using the formula in Table 5.3, taken here from Howell 1987, Chap. 9), to find medium differences (effect size = 0.5 SDs) with a probability of 0.94, and small differences (effect size = 0.2 SDs) with a probability of 0.29. The use of standard deviations as a measure allows this calculation to be unitless and to apply to all differences between models and subjects and also between model conditions. The differences between model conditions here have an effect size of more than 2—in this case, the difference in means of the challenge and threatened model's subtraction attempts divided by the (pooled) standard deviation (54.5–46.8)/3.5, is an effect size of 2.14—so there is more than adequate power to find reliable differences between the model conditions of threat and challenge. Practically, for our model, 100 runs provide more than enough power (0.94) for the example model's effect size of interest (e.g., medium = 0.5).

We suggest that a power of 0.90 for the expected effect size can provide a suggestion of how many runs are required when runs are expensive, and a power of 0.99 when runs are inexpensive. Table 5.4 thus provides a bracketing of number of runs based on a range of power. We choose 0.99, a relatively high number, because model runs are usually inexpensive, and because we wish to understand our model clearly and completely. Table 5.4 provides example values for runs assuming t-tests between means are used. Other values of alpha, other types of measures, and other tests are possible, but other choices for these values do not change the

Table 5.4 The required number of runs (N) to find the given effect sizes (for t tests with α = 0.05, two-tailed) for a range of effect sizes with power = 0.90

Mean effect size	N	δ	Power
0.1	2,178	3.30	0.90
0.2 (Cohen's small)	545	3.30	0.90
0.5 (Cohen's medium)	88	3.30	0.90
0.8 (Cohen's large)	34	3.30	0.90
2.14 (effect reported here)	5	3.30	0.90

This table uses = effect size * sqrt (N/2) (Howell 1987, pp. 201–202, and associated appendix to compute power for the value of δ)

Table 5.5 The required number of runs (N) to find the given effect sizes (for t tests with $\alpha = 0.05$, two-tailed) for a range of effect sizes with power = 0.99

Mean Effect Size	N	δ	Power
0.1	3,528	4.20	0.99
0.2 (Cohen's small)	882	4.20	0.99
0.5 (Cohen's medium)	142	4.20	0.99
0.8 (Cohen's large)	56	4.20	0.99
2.14 (effect reported here)	8	4.28	0.99

This table uses δ = effect size * sqrt (N/2) (Howell 1987, pp. 201–202, and Appendix Power to compute power for the value of δ)

conclusion that increasing the number of runs is desirable to increase power and stabilize the mean and SD, and that power calculations can be used to suggest the number of runs to perform.

Table 5.3 shows the power for a range of effect sizes with 100 runs, which we used here. Table 5.4 provides the number of runs to achieve a power of 0.9 for the same expected effect sizes. This provides a set of reasonable minimum times to run a model where the model runs are expensive.

Table 5.5 provides the number of runs required to achieve a power of 0.99 for the same expected effect sizes. This provides a set of reasonable maximum runs for various effect sizes. If we expected an effect size of 0.8, then 56 runs would provide a power of 0.99 to differentiate predictions from different settings of the model. If we would like to differentiate an effect size between model conditions of 0.2 (Cohen's small effect) then 882 runs would be required for a power of 0.99, and if an effect size of 0.1, then 3,528 runs. This last effect is but 10% of a standard deviation, but if we are interested in that difference, and the model predicts such differences, we can have the statistical power to detect it.

5.6 Discussion and Conclusions

We have presented an example of how many times to run a simulation to understand its predictions. This model's behavior represents theoretical predictions. Therefore, the theory's predictions should be as stable as possible. The results of our example model illustrate that models, where possible, should be run until their predictions are stable. This is particularly important when the model's performance includes predictions of variance in behavior. With a stochastic model implemented as a computer program, we do not wish to sample its behavior, but report its predictions accurately. Thus, we recommend reporting performance based on a larger number of runs than appears to be typically done, and reporting the variance in the predictions. The results shown in Figs. 5.2, 5.3, and 5.4 show that running a model once, or twice, or even several times per human experimental subject, typically will not provide completely accurate predictions and will sometimes provide uncharacteristic predictions.

The power calculations presented here provide a rational way to choose the number of runs. The rationale uses the size of the differences between model conditions and the desired probability of finding these differences to choose the number of model runs to report. This calculation is based on a simple equation included in most introductory statistics books. The calculations are based on standard deviations, which mean that the model's standard deviation or mean does not have to be known before the model is run.

Although we used 100 runs for our serial subtraction model, we recommend 150 runs as a reasonable number that provided very stable predictions for medium to large effects. Power calculations support the use of 150 runs as a useful number for most effect sizes and phenomena of interest to this subtraction model. If one is interested in smaller effect sizes (e.g., Cohen's small effects), then more runs will be required. If one is exploring how a model works or the model runs are expensive, then fewer runs may be appropriate, allowing that there will be less power to see differences between model conditions or between model and data, and a greater likelihood that the predictions are not stable. Other effect sizes and power requirements than those reported here can be used as well.

These results suggest that most of the papers in Table 5.1 did not report stable predictions for their model. While none of the papers in Table 5.1 reported effect sizes per se, large effects are relatively rare, and some of the models were examining what appear to be small to medium effects. On the other hand, the model that was run 7,200 times was almost certainly run enough times, although we agree that if resources are not an issue, then it is best to err on the side of caution.

This use of power analysis particularly helps when model runs are expensive. For example, humans-in-the-loop simulations (e.g., Thiruvengada and Rothrock 2007), models of hour-long experiments that run in real-time (e.g., Schoelles and Gray 2001), models that work with physical robots (Ritter et al. 2007a), or models run over large number of parameter settings (Best et al. 2008; Lovett et al. 2000; Ritter et al. 2009) become difficult to run many times. In these cases, this calculation lets modelers know how many runs are sufficient given the effect size of interest. The power analyses and graphs of the model's output can provide guidance on how many is enough.

These analyses also encourage modelers to think about effect sizes. These are not always known, however, it is useful to consider the effect size of the effect of interest. Where the effect size is small, more subjects need to be run and the model needs to be run more times to get stable predictions. Where the effect is large, less work generally has to be done. This should encourage researchers to look at large effects first.

Would this admonition apply to other models or other aspects of models? Absolutely! These results are not dependent on the specific architecture, but rather on the fact that the predictions have a distribution of outcomes. Soar models that include stochastic elements, for example, Miller and Laird's (1996) categorization model and Soar models with stochastic memory would similarly benefit from multiple runs, and could use the same tables. Psychology experiments already use these types of calculations, or should.

These results would also apply to different statistical tests for different measures, for example, Chi-square on categorical outputs, or different analyses, such as regression, although the power calculations would be different. If the comparison of interest was another measure, such as types of errors, then the percentage and types of errors (which this model makes) become clearer when more of its behavior has been examined. As models become more complex, the number of runs may need to be adjusted because of the additional cost of running the model, however, the cost of running the model additional times is typically much less expensive than not accurately representing and understanding its predictions.

What does this approach not answer? It does not tell you what effect size you will find interesting, or how many times to adjust your model (related to overfitting). It does not tell you what to do if the model does not fit the data; indeed, it suggests that if you run your model long enough, your significance tests will get accurate enough to find even small differences between model and data. These remain interesting and important problems, but at least we can hope that simulations are run enough to be thoroughly understood.

Acknowledgments Earlier versions of this work have been presented at the US Air Force Workshop on ACT-R models of human-system interaction, and ONR workshops on cognitive architectures. Participants there provided useful comments. This project was supported by ONR award N000140310248 and DTRA HDTRA1-09-1-0054. Axel Cleeremans, Andrew Reifers, and Lael Schooler provided comments to improve this paper. The views expressed in this paper do not necessarily reflect the position or the policies of the US Government, and no official endorsement should be inferred.

References

Anderson JR (2007) How can the human mind exist in the physical universe? Oxford University Press, New York, NY

Anderson JR, Lebiere C (1998) The atomic components of thought. Erlbaum, Mahwah, NJ

Ball J, Myers C, Heiberg A, Cooke NJ, Matessa M, Freiman M et al (2010) The Synthetic Teammate Project. Comput Math Organ Sci 16:271–299

Best B, Fincham J, Gluck K, Gunzelmann G, Krusmark MA (2008) Efficient use of large-scale computational resources. In: Proceedings of the seventeenth conference on behavior representation in modeling and simulation. Simulation Interoperability Standards Organization, Orlando, FL, pp 180−181

Byrne MD (2003) Cognitive architecture. In: Jacko J, Sears A (eds) The human–computer interaction handbook: fundamentals, evolving technologies and emerging applications. Erlbaum, Mahwah, NJ, pp 97–117

Cohen J (1988) Statistical power analysis for the behavioral sciences, 2nd edn. Lawrence Erlbaum, Hillsdale, NJ

Grant DA (1962) Testing the null hypothesis and the strategy and tactics of investigating theoretical models. Psychol Rev 69(1):54–61

Howell DC (1987) Statistical methods for psychology, 2nd edn. Duxbury Press, Boston, MA

Kieras DE (2003) Model-based evaluation. In: Jacko J, Sears A (eds) Handbook for human-computer interaction. Erlbaum, Mahwah, NJ, pp 1139–1151

Kieras DE, Wood SD, Meyer DE (1997) Predictive engineering models based on the EPIC architecture for a multimodel high-performance human-computer interaction task. Trans Comput Hum Interact 4(3):230–275

Kirschbaum C, Pirke K-M, Hellhammer DH (1993) The Trier social stress test: a tool for investigating psychobiological stress responses in a laboratory setting. Neuropsychobiology 28:76–81

Laird JE, Newell A, Rosenbloom PS (1987) Soar: an architecture for general intelligence. Artif Intell 33(1):1–64

Lovett MC, Daily LZ, Reder LM (2000) A source activation theory of working memory: cross-task prediction of performance in ACT-R. J Cogn Syst Res 1:99–118

Lovett MC, Schunn C, Lebiere C, Munro P (eds) (2004) Proceedings of the sixth international conference on cognitive modelling, ICCM 2004. Mahwah NJ: Erlbaum. http://www.lrdc.pitt.edu/schunn/ICCM2004/proceedings/schedule.htm

Mathews G (2001) Levels of transaction: a cognitive science framework for operator stress. In: Hancock PA, Desmond PA (eds) Stress, workload, and fatigue. Erlbaum, Mahwah, NJ

Miller CS, Laird JE (1996) Accounting for graded performance within a discrete search framework. Cogn Sci 20:499–537

Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge, MA

Quigley KS, Feldman Barrett LF, Weinstein S (2002) Cardiovascular patterns associated with threat and challenge appraisals: a within-subjects analysis. Psychophysiology 39:292–302

Ritter FE (1988) Extending the Seibel−Soar model: presented at the Soar V workshop held at CMU

Ritter FE (2003) Social processes in validation: Comments on Grant (1962) and Roberts and Pashler (2000). Comments as part of Symposium on Model Fitting and Parameter Estimation. In ACT-R Workshop, 129–130

Ritter FE, Young RM (2001) Embodied models as simulated users: introduction to this special issue on using cognitive models to improve interface design. Int J Hum-Comput Stud 55(1):1–14

Ritter FE, Reifers A, Klein LC, Quigley K, Schoelles MJ (2004) Using cognitive modeling to study behavior moderators: pre-task appraisal and anxiety. In: Proceedings of the human factors and ergonomics society. Human factors and ergonomics society, Santa monica, pp 2121−2125

Ritter FE, Van Rooy D, St Amant R, Simpson K (2006) Providing user models direct access to interfaces: an exploratory study of a simple interface with implications for HRI and HCI. IEEE Trans Sys Man Cybern Part A Sys Hum 36(3):592–601

Ritter FE, Kukreja U, St Amant R (2007a) Including a model of visual processing with a cognitive architecture to model a simple teleoperation task. J Cogn Eng Decis Mak 1(2):121–147

Ritter FE, Reifers AL, Klein LC, Schoelles MJ (2007b) Lessons from defining theories of stress for architectures. In: Gray W (ed) Integrated models of cognitive systems. Oxford University Press, New York, NY, pp 254–262

Ritter FE, Kase SE, Klein LC, Bennett J, Schoelles M (2009) Fitting a model to behavior tells us what changes cognitively when under stress and with caffeine. In: Proceedings of the

biologically inspired cognitive architectures symposium at the AAAI fall symposium. Keynote presentation, Technical report FS-09-01. AAAI Press, Menlo Park, pp 109−115
Schoelles MJ, Gray WD (2001) Argus: a suite of tools for research in complex cognition. Behavior Res Methods Instrum Comput 33(2):130–140
Thiruvengada H, Rothrock L (2007) Time window-based performance measures: a framework to measure team performance in dynamic environments. Cogn Tech Work 9(2):99–108
Tomaka J, Blascovich J, Kelsey RM, Leitten CL (1993) Subjective, physiological, and behavioral effects of threat and challenge appraisal. J Pers Soc Psychol 65(2):248–260
Tomaka J, Blascovich J, Kibler J, Ernst JM (1997) Cognitive and physiological antecedents of threat and challenge appraisal. J Pers Soc Psychol 73(1):63–72

Chapter 6
Training for Metacognition in Simulated Environments

Anand Tharanathan, Paul Derby and Hari Thiruvengada

Abstract Metacognition has been recognized as an important mechanism in the learning process within the cognitive psychology and education literatures. However, due to its focus on relatively static domains, there are several constraints in applying the concept to real-world domains that are highly complex and dynamic in nature. For example, being able to self-regulate the selection of our skills and strategies is essential to maintain a high level of human performance in dynamic environments. Therefore it is important to identify effective training mechanisms to improve metacognition while performing in real-world contexts. An effective platform for cognitive training is human-in-the-loop simulations or virtual environment-training. Hence, in this chapter, we have briefly described the manner in which metacognition is currently defined in the literature and the limitations in its current direction. After identifying the limitations, we provide a definition for the concept of metacognition that may increase its applicability to dynamic domains. Furthermore, we have listed guidelines for developing effective metacognitive training methods in virtual environments as well as an example of the application of these guidelines.

6.1 Metacognition

When interacting in complex and dynamic domains, it is essential for humans to remain attentive to their changing environment. The efficiency of one's cognitive system significantly affects performance in such environments. The self-regulation of our cognitive activity (i.e., metacognition) is important for maintaining a high level of human performance levels, especially in complex and dynamic environments.

A. Tharanathan (✉) · P. Derby · H. Thiruvengada
Honeywell ACS Labs, Golden Valley, Minnesota, USA
e-mail: anand.tharanathan@honeywell.com

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_6,

The term metacognition has been defined as our thoughts about our cognition (Flavell 1979). Specifically, metacognition can be reduced to our declarative knowledge about our cognitive processes (i.e., metacognitive knowledge) as well as the regulation of those cognitive processes (i.e., metacognitive monitoring and control; Cavanaugh and Perlmutter 1982; Schraw 1998). For example, we possess beliefs about our cognitive abilities, knowledge about the nature of the task, as well as knowledge about what cognitive strategies to employ as well as when and how to employ them (Pintrich et al. 2000). This knowledge then affects how we actively regulate our cognitive processes.

Metacognitive processes are not limited to any single facet of cognition. Instead, metacognition has been applied to many cognitive domains such as memory (Dunlosky et al. 2007), comprehension (Maki et al. 2005), strategy selection (Karpicke 2009), recognition (Cohen et al. 1996), and skill combination (Mayer 1998). However, we will couch the following discussion within the realm of learning.

6.1.1 Framework of Metacognition

Nelson and Narens (1990) proposed a framework of metacognition that described the relationship between our cognition and our metacognition. The framework consists of two levels: the object level of cognition and the meta-level of cognition. The object level contains everyday cognitive processes, such as learning, problem-solving, decision making, etc. However, the meta-level contains a model of a person's understanding of the cognitive processes required to perform any given task. This model of understanding contains both pre-existing metacognitive knowledge (e.g., beliefs and ideas about one's learning abilities, past experiences with learning strategies, and the task at hand) and the results of our ongoing assessment of our learning activities (Nelson and Narens 1990).

These two levels interact cyclically through means of monitoring and control (Nelson and Narens 1990). First, metacognitive monitoring is described as the flow of information from the object level to the meta-level. That is, monitoring updates one's mental model in the meta-level based on our judgments about the state of the cognitive processes at the object level. For example, we make judgments of learning (JOL) about the material we are currently attempting to learn. In other words, we make a judgment about the likelihood that we will remember the material later. Accurate JOLs have been found to be positively related with faster learning (Dunlosky et al. 2005).

From the judgments we make about our cognitive behaviors, we update our meta-level model. The new information with our meta-level model is used to influence our current and future cognitive behaviors. This information flow from the meta-level to the object level is referred to as metacognitive control. For example, a high JOL (i.e., monitoring) might result in the termination of practicing or studying (i.e., control). Other metacognitive control mechanisms include the selection of item(s) to learn and selection of the learning strategy (Dunlosky et al. 2007). Consequently, these modified cognitive processes are monitored and the meta-level model is, again, updated.

6.1.2 Self-Regulation of Learning

Adequately monitoring and controlling our cognitive behaviors is only part of what allows us to regulate our learning. The remaining mechanisms have been described within (Zimmerman 2000) model of self-regulated learning. Self-regulated learning refers to the self-motivated, proactive process where "learners transform their mental abilities into academic skills" (Zimmerman 2002). Specifically, self-regulated learning involves using the appropriate knowledge, skills, and attitudes to achieve our learning goals (Zimmerman 2000).

The entire self-regulated learning process is comprised of a three-phase cycle: (1) the forethought phase, (2) the performance phase, and (3) the self-reflection phase (Zimmerman 2000). Upon receiving a learning task, learners engage in the first phase, the forethought phase. Specifically, effective self-regulated learners will begin to set learning goals and decide on the appropriate strategy to reach these goals. In addition, self-regulated learners assess their interest in learning the task, their motivation to complete the task, as well as their perceived ability to complete the task (i.e., self-efficacy).

Given that the learner has chosen a goal and strategies to achieve the goal, the self-regulated learner would enter the second phase, or the performance phase. In this phase, the learner uses his or her metacognitive processes as previously discussed. That is, the learner employs the selected learning strategies and monitors their understanding and learning progress.

The evaluation of the learner's self-monitoring occurs within the third phase. That is, the learner self-reflects about the status of their learning against another standard, such as prior learning performance, others' learning performance, or even a set standard of learning performance (e.g., a rubric). If the current status does not compare well with the standard, an effective self-regulated learner would return to the forethought phase to reevaluate the task goals and perhaps select an alternative learning strategy. However, if the learner evaluated his or her performance meet or exceed the standard, then the learner may choose to discontinue the learning behaviors.

6.1.3 Metacognitive Accuracy

One potential limitation with (Zimmerman's 2000) model of self-regulated learning is the assumption that learners are able to recognize when performance does not align with a standard. In fact, our metacognitive judgments are not always correct. Metacognition has been proposed to be highly subjective (Koriat 1997; Veenman et al. 2002). As briefly discussed, our metacognitive knowledge and ability to self-monitor can be biased by the material being learned (Rawson and Dunlosky 2002), previous experience with the material being learned (Serra and Dunlosky 2005), beliefs about our own abilities (Wagner et al. 1989), and the context and conditions of our cognition (Kelly et al. 2002). Due to the subjective nature of our metacognition, it is believed to be prone to errors (Serra 2006).

Despite any variability in the accuracy of metacognitive judgments, the calibration between these judgments and performance can be improved through feedback. Specifically, feedback about previous performance has a tendency to improve JOL accuracy and test performance over trials (Koriat 1997). In this study, participants were instructed to study a list of items, submit a JOL rating, and then take a test on the same items. After repeating these steps in later trials, both performance and accuracy tended to increase. It had been suggested that participants based their JOLs on their performance in the preceding trials (Finn and Metcalfe 2007; Hertzog et al. 1990; Thiede 1999). Therefore, feedback about previous performance can affect how a learner evaluates the same type of performance in the future.

6.1.4 Measuring Metacognition

There are a variety of means of measuring metacognitive activity. First, the most typical method is eliciting the magnitude of success in performing a cognitive activity. For example, the researcher would ask questions such as, "How well did you learn?" or "How well did you understand?" These judgments are often rated on a 0 (i.e., no understanding or learning) to 100 (i.e., complete understanding or learning) scale (Maki et al. 2005). Second, the magnitude rating could be directly compared with a performance score (Tobias and Everson 2000). This comparison would result in the relative accuracy between the judgment and performance. Third, metacognition can be measured in terms of bias, or one's magnitude of under-confidence or over-confidence about their JOL (Maki 1998). Fourth, metacognitive control can be extracted during elicitation methods such as think-aloud protocol or the use of interviews (Pintrich et al. 2000).

6.1.5 Metacognition in Human-in-the-Loop Simulations

In today's training world, the use of human-in-the-loop simulations (or virtual environments as they are generally termed) for training are extremely popular for several reasons (Shines 2002). First, this popularity is mainly due to the ease associated with developing training scenarios more efficiently and less expensively. Second, simulations allow trainees to be situated within realistic training environments. Because of this, trainees are able to develop and practice cognitive skills (e.g., problem-solving, strategy, recognition, etc.) that can be transferred to the real-world. Lastly, virtual environments negate the problems associated with trainees being located at different geographical locations.

Minimal prior research has looked at metacognitive training in virtual environments despite the indication that training programs developed within virtual and gaming environments often encourage learning to be self-paced (Garris et al. 2002). That is, trainees may have the ability to practice virtual training tasks until they believe they have acquired the skill(s). This requires trainees to self-regulate (e.g., monitor and control) their learning (Ford et al. 1998).

Successful learners will have calibrated their accuracy. However, overconfident learners may have inaccurately assessed their JOL and terminated their learning early, which may have led to poor performance (Osman and Hannafin 1992). One study in particular had examined metacognition within a simulated command and control environment. Fiore et al. (2002) found that poorer metacognitive ability resulted in poorer performance in the simulated environment. Specifically, higher metacognitive bias (e.g., more over-confidence) led to less efficient learning.

Nietfeld et al. (2008) have evaluated several self-regulated learning factors that affect performance within the virtual learning environment, CRYSTAL ISLAND. Specifically, Nietfeld et al. (2008) measured goal orientation, metacognitive monitoring, and situation interest in relation to the participants' performance. Specifically, metacognitive monitoring was positively correlated to the number of completed actions, number of completed goals, and the overall score of the learning exercise. In addition, monitoring was negatively correlated with the number of guesses made about the final outcome of the learning exercise. In other words, better monitoring abilities were related to higher problem-solving accuracy and learning performance within the virtual learning environment.

In summary, although prior research has investigated metacognitive training, there are critical limitations in its current direction. First, metacognition has been primarily studied within static domains (e.g., memory, comprehension, etc.). Therefore, these results may only be applicable to other static environments. Metacognitive training would require further refinement in order to make it applicable to more dynamic domains. Second, research specifically incorporating dynamic virtual environments and metacognitive training is very sparse.

Given the limited prevalence of empirical research concerning metacognition within virtual or gaming environments, there is a need for more empirical research in this area. Specifically, there is a need to improve upon currently existing operational definitions, concepts, and methods of metacognitive training. This should make the applicability of metacognitive training easier and more flexible to real-world contexts that are more dynamic and complex in nature.

6.1.6 Metacognitive Training as a Servo-Mechanism

The objective measurement and evaluation of metacognition and self-regulated learning is not straightforward. In fact, it can be influenced by biases, the task at hand, and self-perceptions (Zimmerman 1995). Moreover, when metacognition has to be measured and evaluated in complex and dynamic domains, the process becomes even more difficult. Consequently, it is important to adopt an effective methodology to validly gauge metacognition. Therefore, we will propose a concept and method to measure and evaluate metacognition in complex and dynamic domains. Then, we will define the specific steps involved in the process. Finally, we will describe how it was applied within a human-in-the-loop simulation for a military task.

6.2 The Concept

A servo-mechanism functions on error-based information processing, adjustment and maintenance. If the system identifies a disturbance, then it tries to correct it and improve overall performance (von Bertalanffy 1968). Metacognitive processes can be compared to this mechanism. Specifically, when trainees gauge (or think about) their own cognition, it is important for them to be able to identify their own cognitive errors and determine methods to overcome those errors.

Importantly, a strategy that streamlines and enhances the efficiency of any system is the categorization of information coming in and leaving the system. Therefore, in a metacognitive training context, if trainees can think about their own cognition in a very categorized (or systematic) manner, it may make it easier for them to detect the errors in their own cognition. So, how can this be made be possible?

One method is to systematically categorize the overall metacognitive training into three main stages of cognition: (1) *information acquisition* (2) *information analysis and* (3) *decision making and action selection*. Each required cognitive skill that has to be trained should be compartmentalized into one or more of these stages. Once the cognitive skills are compartmentalized into one of these three categories, it is more efficient to determine what skills are required to address incoming unprocessed information. In addition, the flow of processed information between the information processing stages and out of the cognitive system would also be more effective. In other words, if trainees start thinking about a task in relation to cognitive skills housed within one or more of the three stages of cognition, they can efficiently train for, apply, and detect errors in the utilization and application of these skills. A schematic representation of this mechanism is shown in Fig. 6.1.

For example, in a military fire team task, it is important for a fire team leader (FTL) to scan an environment and identify hidden areas in a room. Therefore, scanning behavior would be a cognitive skill within the *information acquisition* stage. The next step for a FTL is to analyze the whole situation based on the information gathered by scanning the environment. FTLs have to typically utilize expert processing mechanisms such as cue-based processing. For example, there could be a piece of furniture that inhibits the fire team from scanning the whole room. More importantly, such a piece of furniture could be occluding an explosive device from the view of the fire team members (FTM). The fire team would be required to recognize this potential danger. Hence, cue-based information analysis would be a skill within the *information analysis* stage. The following step for a FTL is to quickly predict the immediate threat involved in the situation and decide

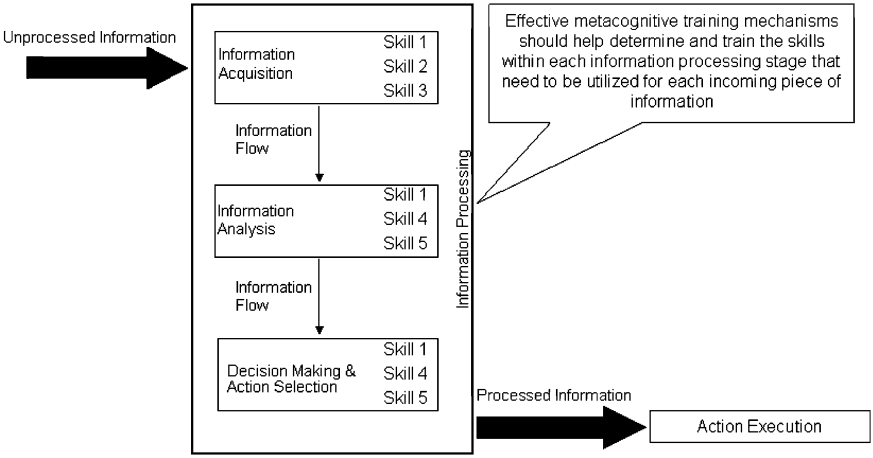


Fig. 6.1 A schematic representation of the cognitive information processing system and information flow with implications for an effective metacognitive training mechanism

whether to pull out of a room, instruct a fire team member to shoot, or even scan an adjoining room. Therefore, threat based (or risk-based) processing of information to arrive at decisions would be a skill within the *decision making and action selection* stage.

Following the execution of the cognitive skills required to complete the tasks being trained, trainees should be given appropriate feedback. First, it is important to list out the cognitive skills within each stage of information processing that trainees need to utilize. Second, the trainees should be trained on these specific skills and self-assess how well they are progressing. Third, feedback should be given on their overall performance as well as explain how each skill is housed within each of the three stages of information processing. This will allow trainees to assess how their metacognitive judgments align with the actual state of their performance. If the trainees self-perception is accurate, then there should be no difference between the trainee's actual performance and self-rated (metacognitive) performance. In contrast, if there is a difference between the two (i.e., an error), then, with proper skill-based and information processing stage-based metacognitive training the trainees should be able to identify strategies to negate the errors. Thus, the proposed metacognitive training mechanism is similar to a servo-mechanism.

6.3 Steps Involved

While developing efficient metacognitive training modules that are applicable to real-world contexts, the overall process can be quite rigorous. Here, we have laid out five phases involved in the development of an effective metacognitive training module. We have kept the process as domain-independent as possible. Therefore, for a specific domain, minor modifications have to be done.

Phase 1: Understand the domain and the task

Step 1: work with subject matter experts (SMEs) to identify specific skills associated with a work domain.

Step 2: with hierarchical task analysis and knowledge elicitation techniques break down the tasks of a particular person.

Step 3: once the tasks are broken down, identify the cognitive skills required for the task.

Phase 2: Compartmentalize the cognitive skills within the information processing stages

Step 1: map and categorize the identified cognitive skills into the three stages of cognition.

Step 2: differentiate the weights of each cognitive skill within each stage of information processing. That is, identify what cognitive skills are more important than others in the completion of a task.

Phase 3: Developing the measurement and feedback mechanism

Step 1: identify the metacognitive probes, or questions, that should be administered during the measurement of metacognitive efficiency. These questions may assess how well the trainee believes that he or she has performed or learned, etc. Alternatively, questions could assess how efficiently the trainee had completed each stage of cognition (i.e., information acquisition, information analysis, and decision making).

Step 2: map each probe to specific skills within the stages of information processing.

Step 3: a data reconciliation engine should compute the overall accuracy of trainees' ratings. If the analysis based on the ground truths and trainees' ratings indicates a difference between these two parameters, then it indicates that there is an error in a trainee's metacognitive judgments.

Phase 4: Delivering the training module

Step 1: based on the discrepancies between the trainees' actual performance and trainees' metacognitive judgments, training modules should be developed. These modules should have tailor-made suggestions for each trainee and these suggestions should be mapped on to a specific cognitive skill within an information processing stage.

Step 2: the training module should reflect the associated weights of each cognitive skill. For example, if a skill is more critical within an information processing stage, then that skill has to be given a higher focus in the training module.

Phase 5: Continuously improving the training module

Step 1: continuously engage trainees in modules that aid them in identifying their errors, determining mechanisms to correct them, and eventually becoming error-free. Such an approach would encourage trainees to think about a task more systematically and in a categorized manner.

Step 2: continuously improve the training module based on lessons learned from actual application in the field. The improvements may be the addition of specific skills within an information processing stage, different weights for already existing skills etc.

Step 4: validate the additions made to the training module with SMEs.

The steps mentioned in this section can also be considered as heuristics or "rules of thumb" rather than strictly time-ordered steps.

6.4 Application of the Concept within a Human-in-the-Loop-Simulation

Current training systems have critical limitations such as poor adaptability, lack of real-time assessment engines, deployment issues and the need for manual intervention while training (e.g., Thiruvengada et al., this chapter). To overcome some of these limitations, we developed the performance feedback engine for conflict training (PerFECT). This engine was designed to place trainees in appropriate virtual environment training scenarios with apt time compression to focus on critical skills; to collect performance metrics based on cognitive, behavioral, environmental and human interaction models; to provide relevant and timely feedback to the trainee; and to automatically present novel training scenarios based on identified skill deficiencies (Thiruvengada et al. this chapter).

PerFECT is a training module administered within a virtually simulated environment to train and evaluate a FTLs cognitive decision making at the level of a small unit tactical mission. In this task, each trainee was a part of a fire team consisting of a fire team leader (FTL) and three fire team members (FTM). As an FTL, the trainee was responsible for entering and clearing a series of rooms within an urban building.

After multiple rigorous interviewing and knowledge elicitation sessions with subject matter experts, we identified that the specific skills required for an FTL while entering and clearing a room are *movement*, *tactical*, *communication*, *technical* and *reporting* skills. It is important to note that there are a variety of tasks that comprise a skill. For example, a task could be moving to a stack position, entering a room, or throwing a grenade. The PerFECT training system was designed to measure, evaluate, provide feedback, and train for all five skills. For a detailed review of the PerFECT system and architecture, the readers are advised to refer to Thiruvengada et al. (this chapter). Here we will focus only on how the feedback mechanism for metacognitive training was administered.

6.4.1 The Feedback Engine for Metacognitive Training

There are two feedback clients: one for the trainee(s) and one for the trainer. The trainer's feedback client (Fig. 6.2) allows the trainer to observe the trainees'

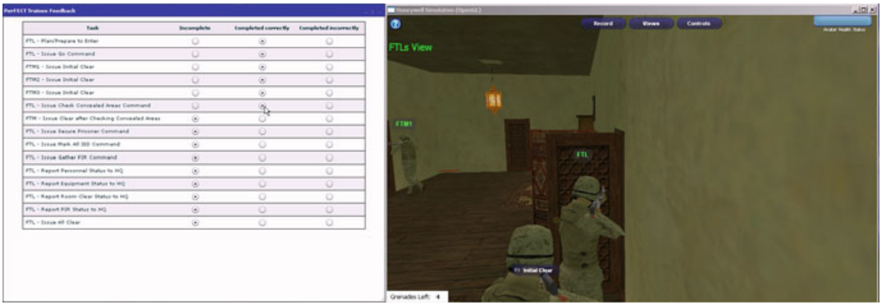


Fig. 6.2 Trainer version of feedback client allows trainers to evaluate communication tasks in real time

behaviors within the simulated virtual environment. It also allows the trainer to make notes about the trainee's performance or identify the tasks that the trainee had accomplished along the way. Simultaneously, the system also records the trainee's actions automatically.

At the end of each scenario, the trainee's feedback client allows the trainee to rate his or her own performance for each skill on a continuum as being "trained," "needs practice," or "untrained". Trainees also need to provide a rationale and justification for their ratings. In addition, the trainer rates the trainee performance for each skill along the same continuum. The trainer can also provide any written comments about how the trainee can improve his or her performance.

Trainees receive overall feedback on both how they actually performed and on their self-rating of their performance (i.e., metacognitive understanding) of their performance of all five skills (see, Fig. 6.3). Ideally, there should be minimal difference between the trainee's actual and metacognitive performance.

In addition, as shown in Fig. 6.4, trainees are able to look at their performance on each skill (e.g., *movement*, *tactical*, etc.) within more detail. Specifically, the trainees are able to see how they have performed over time– in terms of both actual performance and self-rated (metacognitive) performance.

Thiruvengada et al. (this chapter) described a pilot test that used the feedback engine for metacognitive training. The pilot test indicated that the trainees' ($N = 2$) performance improved over time with the use of the feedback mechanism. After the pilot test, participants indicated that this feedback mechanism that compared their judgment of their performance and their actual performance helped them monitor and control their tasks more effectively. This feedback mechanism also allowed the trainees to calibrate their self-assessments, which helped to correct issues related to over-confidence and under-confidence. However, a follow-up study with a larger sample size is required to determine the robustness of our results.

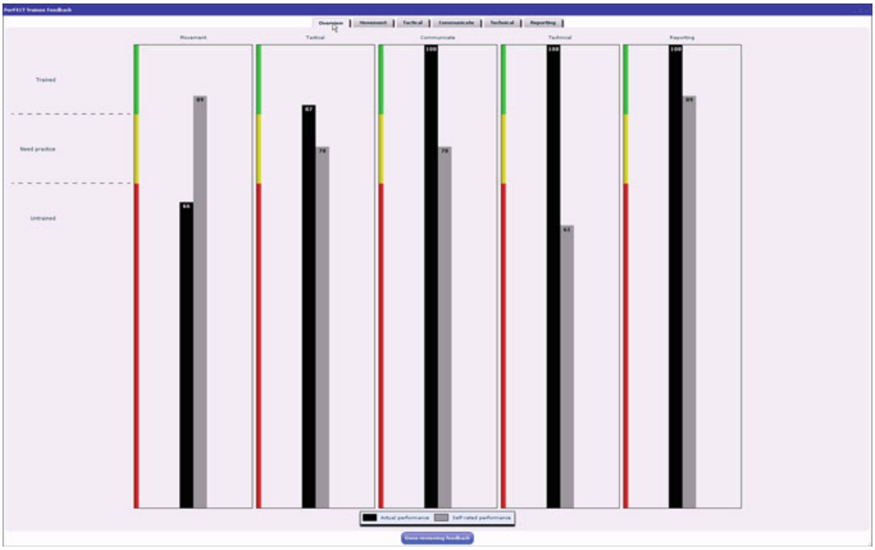


Fig. 6.3 Overall performance feedback that displays actual performance beside metacognitive rating for the five skills included in the training

Fig. 6.4 Detailed feedback on task performance specific to each trained skill

6.5 Future Directions

Given the increasing popularity of self-regulated, human-in-the-loop learning simulations, there is a need for more future research concerning metacognitive processes within these environments. Specifically, researchers should seek to understand how self-regulated learning occurs within real-time, dynamic environments (e.g., training simulations) opposed to strictly static environments (e.g., text comprehension). In addition, research should be conducted to understand how metacognitive processes related to self-regulated learning progress over time with practice.

Several exploratory papers have sought to identify the role of feedback in self-regulated learning (e.g., Butler and Winne 1995; Nicol and Macfarlane-Dick 2006). However these examples provided an assessment of feedback and self-regulated learning in mostly classroom environments. Future research should also assess how feedback affects self-regulated learning in real-time environments. Specifically, researchers should attempt to understand what types of feedback are most effective. In addition, it would be important to understand how the need for feedback changes as the trainee progresses toward expertise.

Lastly, some effort should be taken to improve the availability of embedded measurement within human-in-the-loop simulations. Currently, many metacognition experiments involve querying participants to make judgments about their metacognitive processes (e.g., JOL). Despite the wealth of knowledge accumulated by using queries, this method has been speculated to be highly affected by context. For example, Meshkati et al. (1995) warned that subjective ratings can be influenced by other memories, experiences, attitudes, and current states (e.g., fatigue). Therefore, there is a need to develop and validate means of extracting metacognitive control processes manifested within task performance. For example, one type of metacognitive control can be measured by detecting the amount of time a participant choose to study and learn material (Dunlosky et al. 2007).

References

Butler DL, Winne PH (1995) Feedback and self-regulated learning: a theoretical synthesis. Rev Educ Res 65:245–281

Cavanaugh JC, Perlmutter M (1982) Metamemory: a critical examination. Child Dev 53:11–28

Cohen M, Freeman J, Wolf S (1996) Metarecognition in time-stressed decision making: recognizing, critiquing and correcting. Hum Factors 38:206–219

Dunlosky J, Hertzog C, Kennedy MRF, Thiede KW (2005) The self-monitoring approach for effective learning. Int J Cogn Technol 10:4–11

Dunlosky BJ, Serra MJ, Baker JMC (2007) Metamemory. In: Durso FT, Nickerson RS, Dumais ST, Lewandowsky S, Perfect T (eds) Handbook of applied cognition, 2nd edn. Wiley, West Sussex

Finn B, Metcalfe J (2007) The role of memory for past test in the underconfidence with practice effect. J Exp Psychol Learn Mem Cog 33:238–244

Fiore SM, Cuevas HM, Scielzo S, Salas E (2002) Training individuals for distributed teams: problem solving assessment for distributed mission research. Comput Hum Behav 18:125–140

Flavell JH (1979) Metacognition and cognitive monitoring: a new area of cognitive-developmental inquiry. Am Psychol 34:906–911

Ford KJ, Smith EM, Weissbein DA, Gully SM, Salas E (1998) Relationships of goal orientation metacognitive ability and practice strategies with learning outcomes and transfer. J Appl Psychol 83:218–233

Garris R, Ahlers R, Driskell JE (2002) Games, motivation and learning: a research and practice model. Simul Gaming 33:441–467

Hertzog C, Dixon RA, Hultsch DF (1990) Relationships between metamemory, memory predictions, and memory task performance in adults. Psychol Aging 5:215–227

Karpicke JD (2009) Metacognitive control and strategy selection: deciding to practice retrieval during learning. J Exp Psy Gen 138:469–486

Kelly A, Carroll M, Mazzoni G (2002) Metamemory and reality monitoring. Appl Cogn Psychol 16:407–428

Koriat A (1997) Monitoring one's own knowledge during study: a cue-utilization approach to judgments of learning. J Exp Psychol Gen 126:349–370

Maki RH (1998) Text predictions over text material. In: Hacker DJ, Dunlosky J, Graesser AC (eds) Metacognition in educational theory and practice. Erlbaum, Mahwah

Maki RH, Shields M, Wheeler AE, Zacchilli TL (2005) Individual differences in absolute and relative metacomprehension accuracy. J Educ Psychol 97:723–731

Mayer RE (1998) Cognitive, metacognitive, and motivational aspects of problem solving. Instr Sci 26:49–63

Meshkati N, Hancock PA, Rahimi M, Suzanne MD (1995) Techniques in mental workload assessment. In: Wilson EN, Corlett JR (eds) Evaluation of human work: a practical ergonomics methodology, 2nd edn. Taylor and Francis, London

Nelson TO, Narens L (1990) Metamemory: a theoretical framework and new findings. In: Bower GH (ed) The psychology of learning and motivation. Academic Press, San Diego

Nicol DJ, Macfarlane-Dick D (2006) Formative assessment and self-regulated learning: a model and sevel principles of good feedback practice. Stud High Educ 31:199–218

Nietfeld JL, Hoffman KL, McQuiggan SW, Lester J (2008) Self-regulated learning in a narrative entered learning environment. Annual Meeting of Ed-Media, Vienna, Austria

Osman ME, Hannafin MJ (1992) Metacognition research and theory: analysis and implications for instructional design. Educ Tech Res Dev 40:83–99

Pintrich PR, Wolters C, Baxter G (2000) Assessing metacognition and self-regulated learning. In: Schraw G, Impara JC (eds) Issues in the measurement of metacognition. Buros Institute of Mental Measurements, Lincoln NE

Rawson KA, Dunlosky J (2002) Are performance predictions for text based on ease of processing? J Exp Psychol Learn Mem Cog 28:69–80

Schraw G (1998) Promoting general metacognitive awareness. Instr Sci 26:113–125

Serra MJ (2006) Is metacomprehension for multimedia presentations different than for text alone? Unpublished doctoral dissertation. Kent State University, Kent, OH

Serra MJ, Dunlosky J (2005) Does retrieval fluency contribute to the underconfidence-with-practice effect? J Exp Psychol Learn Mem Cogn 21:1258–1266

Shines DP (2002) The name of the game is training: leveraging army gaming to improve training. Armor 450:22–24

Tobias S, Everson H (2002) Knowing what you know and what you don't: further research on metacognitive knowledge monitoring. College Board Report No. 2002-3. College Board, NY

Thiede KW (1999) The importance of accurate monitoring and effective self-regulation during multitrial learning. Psychon B Rev 6:662–667

Veenman MVJ, Prins FJ, Elshout JJ (2002) Initial inductive learning in a complex computer simulated environment: the role of metacognitive skills and intellectual ability. Comput Hum Behav 18:327–341 von Bertalanffy L (1968) General systems theory: foundations development and applications. George Braziller, New York

Wagner DA, Spratt JE, Gal I, Paris S (1989) Reading and believing: Beliefs attributions and reading achievement among Moroccan school children. J Educ Psychol 81:283–293

Zimmerman BJ (1995) Self-regulation involves more than metacognition: a social cognitive perspective. Educ Psychol 30:217–221

Zimmerman BJ (2000) Attaining self-regulation: a social cognitive perspective. In: Boekaerts M, Pintrich P, Zeidner M (eds) Handbook of self-regulation. Academic Press, Burlington

Zimmerman BJ (2002) Becoming a self-regulated learner: An overview. Theor Pract 41:64–70

Chapter 7
Trade Space Exploration: Assessing the Benefits of Putting Designers "Back-in-the-Loop" during Engineering Optimization

Timothy W. Simpson, Dan Carlsen, Matthew Malone and Joshua Kollat

Abstract Trade space exploration is a promising decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving engineering design optimization problems. This is achieved by combining multi-dimensional data visualization techniques with visual steering commands to allow designers to "steer" the optimization process while searching for the best, or Pareto optimal, designs. After introducing the trade space exploration paradigm and visual steering capabilities that we developed, we compare the performance of different combinations of visual steering commands implemented by two users to a multi-objective genetic algorithm executed "blindly" on the same problem with no human intervention. The results indicate that the visual steering commands—regardless of the order and combination in which they are invoked—provide a 4–7× increase in the number of Pareto solutions obtained for a given number of function evaluations when the human is "in-the-loop" during the optimization process. As such, this study provides empirical evidence of the benefits of interactive visualization-based strategies to support engineering design optimization and decision-making. Future work is also discussed.

7.1 Introduction

Many engineering designers employ optimization-based tools and approaches to help them make decisions particularly during the design of complex systems such as automobiles, aircraft, and spacecraft, which require trade-offs between conflicting and competing objectives. Trade space exploration is a promising

T. W. Simpson (✉) · D. Carlsen · M. Malone · J. Kollat
Mechanical & Nuclear Engineering and Industrial & Manufacturing Engineering, The Pennsylvania State University, University Park, PA 16802, USA
e-mail: tws8@psu.edu

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_7, © Springer-Verlag London Limited 2011

alternative decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving design optimization problems. Trade space exploration is an embodiment of the Design by Shopping paradigm advocated by Balling (1999): designers, like consumers, want to "shop" to gain intuition about trades, what is feasible and what is not, and to learn about their alternatives first before making decisions. Balling noted that the traditional optimization-based design process of "(1) formulate the design problem, (2) obtain/develop analysis models, and (3) execute an optimization algorithm" often leaves designers unsatisfied with their results because the problem is usually improperly formulated: "the objectives and constraints used in optimization were not what the owners and stakeholders really wanted… in many cases, people do not know what they really want until they see some designs" (Balling 1999). Similar findings have been reported in other fields. For instance, Wilson and Schooler (1991) have shown that people do worse at some decision tasks when asked to analyze the reasons for their preferences or evaluate all the attributes of their choices. Likewise, Shanteau (1992) observed that when people are dissatisfied with the results of a rational decision-making process, they often change their ratings to achieve their desired result.

This chapter presents results from research that seeks to formalize methods, tools, and procedures to support the trade space exploration process. In particular, we empirically assess the performance of visual steering commands—visually specified controls that allow designers to "steer" an optimization algorithm—developed to support trade space exploration (Simpson et al. 2007). This is achieved by comparing the performance of two users employing different combinations of the visual steering commands to a multi-objective genetic algorithm executed "blindly" on the same problem with no human intervention. Related research in computational steering is discussed next before reviewing the visual steering commands available in our multi-dimensional data visualization software in Sect. 7.3. Section 7.4 describes the test problem used in this work and the experimental set-up for our study. The results and findings are discussed in Sect. 7.5, and future work is outlined in Sect. 7.6.

7.2 Review of Related Work

In the visualization community, interactive optimization-based methods fall primarily into the area of computational steering whereby users (e.g., designers) interact with a model or simulation during the optimization process to help "steer" the search process toward what looks like an optimal solution. The designer observes some sort of a visual representation of the optimization process and then uses intuition, heuristics, and/or some other methods to adjust the search to move toward a design that may not have been intuitive at the beginning of the process. For instance, Wright et al. (2000) applied computational steering to design the geometry and select the grade of glass for a furnace. Kesavadas and Sudhir (2000) created large-scale manufacturing simulations by allowing users to make quick changes “on-the-fly” and continue with the simulation. Messac and Chen (2000) proposed an interactive visualization method wherein the progress of the optimization is visualized—but not steered—throughout the process. Finally, Visual Design Steering (Winer and Bloebaum 2001, 2002) allows users to stop and redirect the optimization process to improve the solution; however, their visualization capabilities are currently limited to 2D and 3D representations of constraints and objectives.

Scott et al. (2002) recently proposed that including humans “in the loop” throughout the decision-making process improves the outcome. They investigated the effects of integrating humans into the optimization process and found that “combining the human’s superior intelligence with the computer’s superior computational speed can result in better solutions than either could produce alone”. Additional advantages include learning about the problem and the interrelationships between objectives and having the ability to guide the solution process in a desired direction and possibly even changing one’s mind while learning (Miettinen and Makela 2006). Solutions generated through human interaction are better understood by the user than solutions merely given to them by an optimization algorithm. Moreover, the computational costs can be significantly reduced since only solutions of interest to the decision-maker are generated (Scott et al. 2002).

Madar et al. (2005) are investigating the effects of human interaction on a particular optimization algorithm, namely, particle swarm optimization. By using their visual, cognitive, and strategic abilities, human users can improve the performance of the computer search algorithm. Thus, interactive optimization approaches seek to combine expert knowledge with computational power. Michalek and Papalambros (2002) propose in their work on architectural layouts that “the designer’s interaction causes the program to dynamically change the optimization representation on-the-fly by adding, deleting, and modifying objectives, constraints, and structural units”. Their “on-the-fly” methodology is applicable for architectural design because of its subjective nature, but the usefulness of it in complex system design conceptualization requires further exploration.

7.3 Our Visualization Software and Visual Steering Tools

To support trade space exploration, researchers at the Applied Research Laboratory (ARL) and Penn State have developed the ARL Trade Space Visualizer (ATSV) (Stump et al. 2004a, b), a Java-based application capable of visualizing multi-dimensional trade spaces using glyph, 1D and 2D histogram, 2D scatter, scatter matrix, and parallel coordinate plots, linked views (Buja et al. 1991), and brushing—a technique used to mask data from a display if it does not meet some filter criteria (Becker and Cleveland 1987). Figure 7.1 shows several examples of

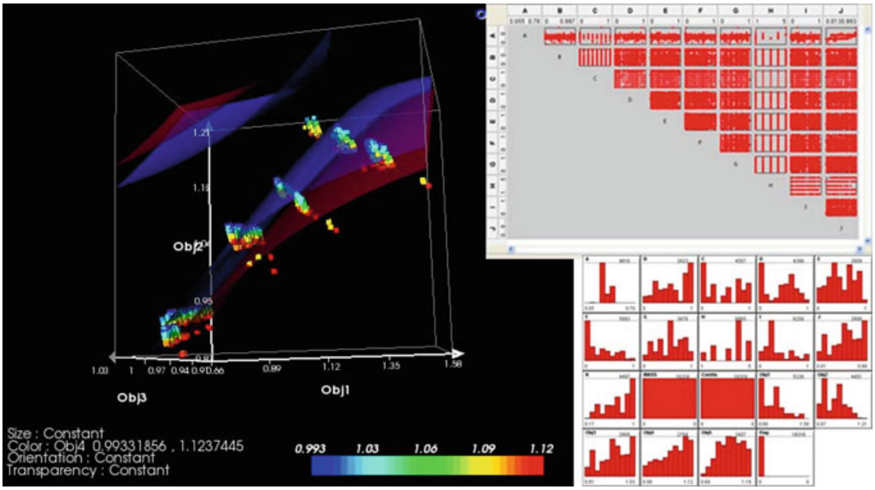


Fig. 7.1 Three displays of data in ATSV: glyph plot (*left*) using color and *x*, *y*, *z* coordinates to display data; scatter matrix (*top right*) showing all two-factor interactions; and histogram (*bottom right*) showing frequency distributions of data that has been generated (inputs and outputs)

ATSV's data visualization capabilities. The glyph plot (left) can display up to seven dimensions by assigning variables to the *x*-axis, *y*-axis, *z*-axis, position, size, color, orientation, and transparency of the icons (e.g., spheres, cubes, stars) used in the glyph plot. In this glyph plot, Obj1, Obj2, and Obj3 from the vehicle configuration problem used in Sect. 7.4 are plotted on the *x*-, *y*-, and *z*-axes, respectively, and the glyphs (cubes) use a color map to indicate the values of Obj4 as noted in the scale at the bottom of the plot; meanwhile, the contours show different values of curb weight for the vehicle. The scatter matrix (top right), a grid of all 2D scatter plots, is useful for visualizing trends and two-way interactions in the data. Histograms (bottom right) show the distribution of the samples in each dimension.

The design variable (input) and performance (output) data for different design alternatives can either be generated offline and then input into ATSV for visualization and manipulation or it can be generated dynamically "on-the-fly" by linking a simulation model directly with ATSV using its Exploration Engine capability (Simpson et al. 2007). If the simulation model is too computationally expensive to be executed in real-time, then low-fidelity metamodels can be constructed and used as approximations for quickly searching the trade space (Wang and Shan 2007). Once this link is in place, ATSV provides a suite of controls to help designers navigate and explore the trade space, including visual steering commands to: (1) randomly sample the design space, (2) search near a point of interest, (3) search in a direction of preference, or (4) search for the Pareto front (Simpson et al. 2007). A brief summary of each follows.

(1) *Design space samplers* are used to populate the trade space and are typically invoked if there is no initial data available. The user can sample the design space

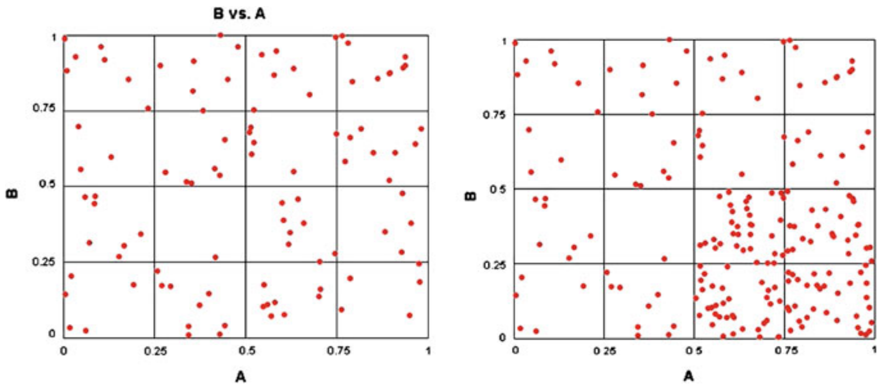


Fig. 7.2 Design space sampler: initially 100 points are randomly generated in the whole space (*left*) and then 100 additional points are generated in the region, $A \geq 0.5$, and $B \leq 0.5$ (*right*)

manually using slider bar controls for each input dimension or randomly. When sampling randomly, the user specifies the number of samples to be generated and the bounds of the multi-dimensional hypercube of X. Monte Carlo sampling then randomly samples the inputs—drawing from a uniform, normal, or triangular distribution—and executes the simulation model, storing the corresponding output in the database. The bounds of the design variables can be reduced at any point to bias the samples in a given region if desired. Figure 7.2 shows an example based on the vehicle configuration problem used in Sect. 7.4. In particular, the designer wished to generate an additional 100 samples in the region $A \geq 0.5$ and $B \leq 0.5$ (right) after viewing the results from the first 100 random samples (left); A and B are inputs to the problem, and the designer simply adjusted the corresponding slider bars in ATSV for these two variables before executing the design space sampler the second time.

(2) *Point samplers*, also referred to as attractors, are used to generate new sample points near a user-specified location in the trade space. The attractor is specified in the ATSV interface with a graphical icon that identifies an n-dimensional point in the trade space, and then new sample points are generated near the attractor—or as close as they can get to it. Unbeknownst to the user, the attractor generates new points using the differential evolution (DE) algorithm (Price et al. 2005), which assess the fitness of each new sample based on the normalized Euclidean distance to the attractor. The normalized Euclidean distance is used to avoid problems that would arise if the magnitude of the objectives varied widely. As the population evolves in DE, the samples get closer and closer to the attractor. An example is shown in Fig. 7.3 where the designer used an attractor to sample in the region where Obj1 is low and Obj3 is high, and ATSV generated additional samples as it explored the trade space in this region, filling in the "gap" that initially appeared in the plot.

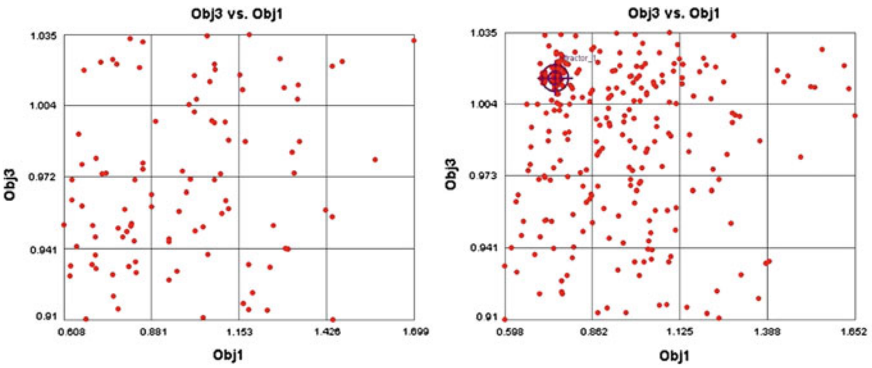


Fig. 7.3 Example of attractor used to explore a region in the trade space that initially had a "gap" after 100 points were generated (*left*); final points cluster tightly around the attractor (*right*)

(3) *Preference-based samplers* allow users to populate the trade space in regions that perform well with respect to a user-defined preference function. New sample points are also generated by the DE algorithm, but the fitness of each sample is defined by the user's preference structure instead of the Euclidean distance. An example of the preference-based sampler is shown in Fig. 7.4. Using ATSVs brushing and preference controls, the user specifies a desire to minimize Obj1 and maximize Obj3 with equal weighting based on the problem formulation given in Sect. 7.4. The initial samples are shaded based on this preference, and then samples are generated, increasing in the direction of preference, namely, the highlighted region of the plot. Using this sampler, the designer now has several good alternatives (i.e., data points) that minimize Obj1 and maximize Obj3.

(4) *Pareto samplers* are used to bias the sampling of new designs in search of the Pareto front once the user has defined his/her preferences on the objectives. The DE algorithm is again used to accomplish this sampling but is modified to solve multi-objective problems (Robic and Filipic 2005). An example of this sampler is shown in Fig. 7.5. Using the same preference (i.e., minimize Obj1 and maximize Obj3 with equal weighting), the initial Pareto front is expanded significantly by using the sampler to find more designs that satisfy this preference. The result is a range of potential solutions that minimize Obj1 and maximize Obj3 for the problem.

These visual steering commands can be used together in any combination to explore the trade space. When used in concert with the ATSV, designers have a powerful multi-dimensional data visualization tool with the capability to "steer" the optimization process while navigating the trade space to find the best design. To determine the extent to which these visual steering commands are effective in

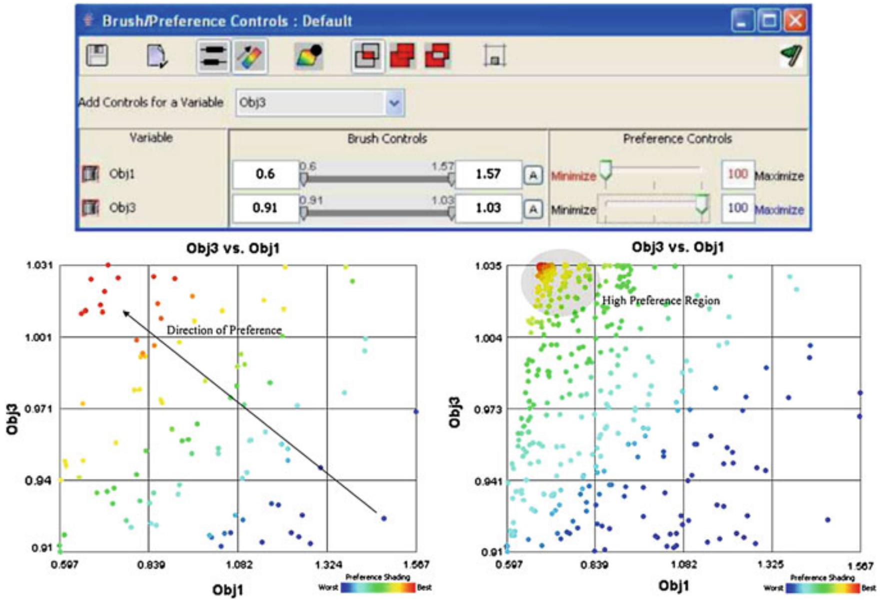


Fig. 7.4 Example of preference-based sampler: designer specifies preferences on Obj1 and Obj3 using the Brush/Preference Controls in ATSV, then samples are generated in the preferred direction, as indicated the arrow (*bottom left*), yielding more designs in the region of high preference

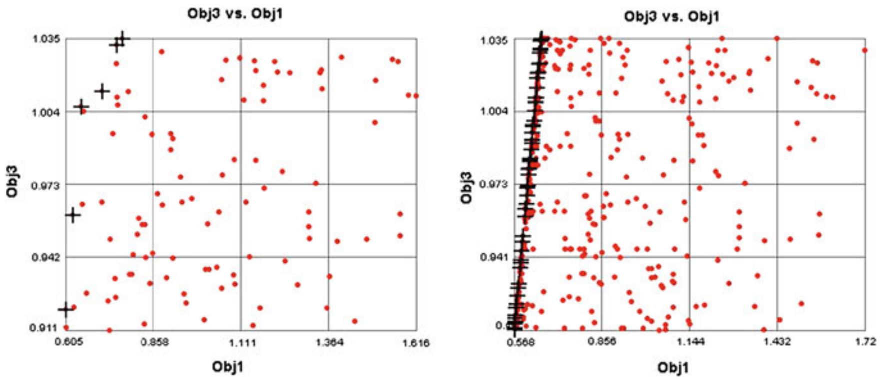


Fig. 7.5 Example of Pareto sampler, which is used to generate points more along the Pareto front as indicated by the +, reflecting the designer's preference to minimize Obj1 and maximize Obj3

locating good designs, the following section describes a study that compares different combinations of these visual steering commands to a multi-objective genetic algorithm executed on the same problem with no human intervention.

7.4 Experimental Set-Up and User Trials

7.4.1 Description of Test Problem

The test problem used in this study is a vehicle configuration model developed to evaluate the technical feasibility of new vehicle concepts (Donndelinger et al. 2006; Ferguson et al. 2005a, b). Table 7.1 summarizes the problem definition used for this trade space exploration example. The inputs to the model are 11 high-level vehicle design parameters: ten continuous variables that define the overall exterior dimensions and positions of the occupants, and one discrete variable, H, that defines the vehicle's powertrain as being one of six options. There are seven outputs from the model, including five measures of performance, vehicle mass, and total constraint violation, which is zero when all the constraints internal to the model are satisfied (i.e., ConVio = 0). The continuous design variables are normalized to [0,1] based on the input bounds while the objectives and vehicle mass are scaled against the baseline. As noted in the table, Obj1 should be smaller than the baseline value while larger values are better for the other four objectives. While stating these very general preferences beforehand may seem counterintuitive to trade space exploration, the end goal here is to demonstrate that the visual steering commands used in conjunction with ATSV are more effective at obtaining an equally desirable Pareto front than by simply allowing a multi-objective genetic algorithm (MOGA) to run "blindly".

7.4.2 Description of User Trials

Two sets of user trials were defined for the study based on the allocated number of function evaluations that could be used: ~5,000 and ~10,000, and two users performed each set of trials to account for any randomness in the algorithms, placement of attractors, or specification of brush/preferences controls. While there are nearly an infinite number of combinations of brushing, preference controls, and visual steering commands that could be implemented in ATSV, we allowed a more experienced user to step through a process that felt "natural" and then had the less experienced user replicate those steps as accurately as possible. The experienced user was asked to do this multiple times, creating four different combinations (Trials 1–4) that each used ~5,000 function evaluations and four different combinations (Trials 5–8) that each used ~10,000 function evaluations.

The ATSV set-up and parameter settings for Trials 1–4 are shown in Fig. 7.6. The appendix describes the specific combinations of brush/preference controls and visual steering commands used for each of Trials 1–4.

All four trials begin with a relatively small set of randomly generated samples before proceeding to different combinations of samplers. The motivation for

Table 7.1 Definition of inputs and outputs to vehicle configuration problem: inputs are normalized against lower and upper bounds while outputs are scaled against the baseline model

Model inputs			
Variable	Type	Lower bound	Upper bound
A	Continuous	0	1
B	Continuous	0	1
C	Continuous	0	1
D	Continuous	0	1
E	Continuous	0	1
F	Continuous	0	1
G	Continuous	0	1
H	Discrete	1	6
I	Continuous	0	1
J	Continuous	0	1
K	Continuous	0	1
Model outputs			
ConVio		0 → feasible	>0 → infeasible
Mass		Baseline = 1	Defines weight class
Obj1		Baseline = 1	Smaller is better
Obj2		Baseline = 1	Larger is better
Obj3		Baseline = 1	Larger is better
Obj4		Baseline = 1	Larger is better
Obj5		Baseline = 1	Larger is better

setting attractors comes from the fact that many designers use pair-wise comparisons in making decisions (Dym et al. 2006); comparing only two objectives makes it easy to see the relationships among them. Not all attractors were placed for this reason; others were placed in an attempt to fill in gaps [similar to how the Gap Analyzer was used (Ferguson et al. 2005b)] in the Pareto front, or to push the Pareto front toward optimality as the trade space exploration process unfolded. The preference and Pareto samplers were also used in an attempt to fill in the Pareto front. Unless specified, the Exploration Engine settings (see bottom left of Fig. 7.6) were set at the defaults of generation size = 25, population limit = 500, and the Best1Bin selection strategy. Figure 7.7 shows the Pareto fronts obtained by both users after performing Trial 4.

The second set of four trials (Trials 5–8) each used approximately 10,000 points, doubling the number of function evaluations allocated to the user. These four trials all began with a small set of random samples to allow the user to specify preferences (see Fig. 7.8), but they then varied widely in the order and type of attractors and samplers used. The appendix describes the user preference settings and specific combination of visual steering commands used for Trials 5–8. Note that these trials also set a preference on ConVio to minimize it before generating too many points, with the exception of Trial 5, which set it halfway through the trial. Unless specified, the same options and parameter settings were used for these

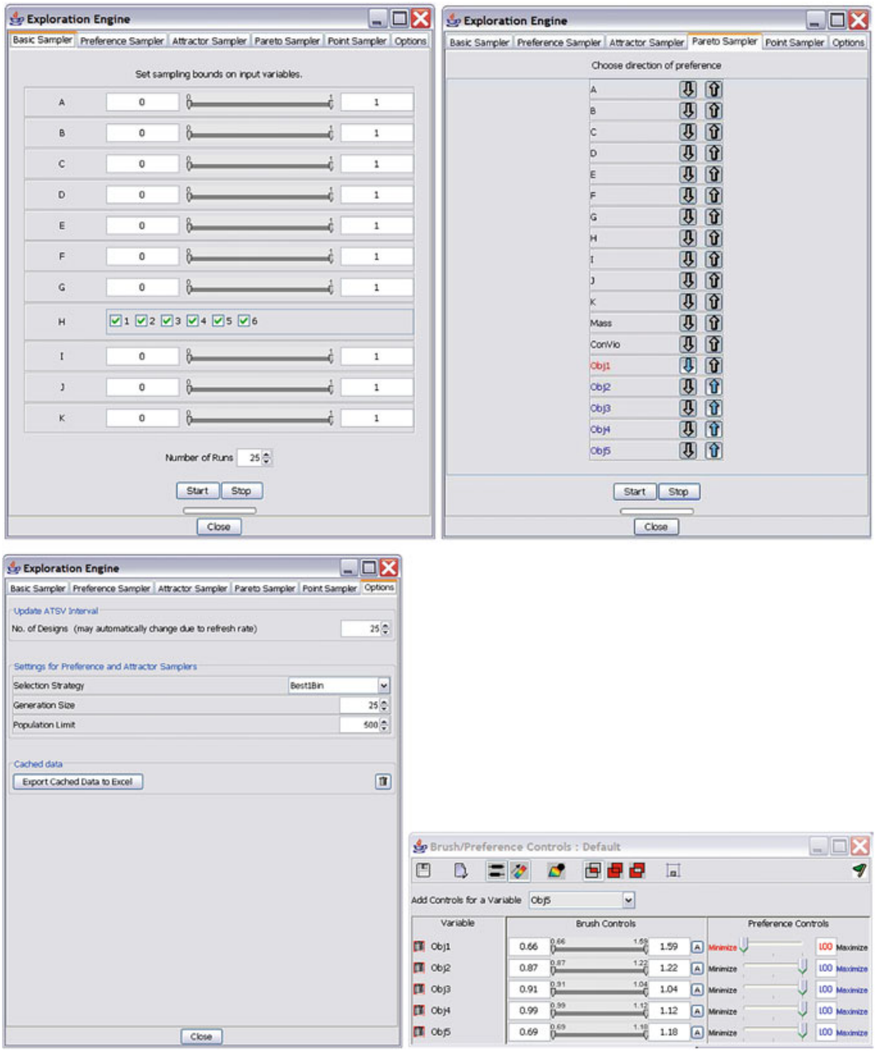


Fig. 7.6 General settings in ATSV for Trials 1–4: basic sampler (*top left*), preferences for Pareto sampler (*top right*), parameter settings for DE search algorithm (*bottom left*), and brush/ preference control settings (*bottom right*)

trials as Trials 1–4 (see Fig. 7.6). Figure 7.9 shows an example of the Pareto fronts that the two users obtained after completing Trial 6.

In summary, the major attributes of each of the trials are as follows:

- Trial 1: Attractors placed based on 2 objective interactions.
- Trial 2: Tried to advance the Pareto front based on what was visible.
- Trial 3: Attractors placed based on 3 objective interactions.

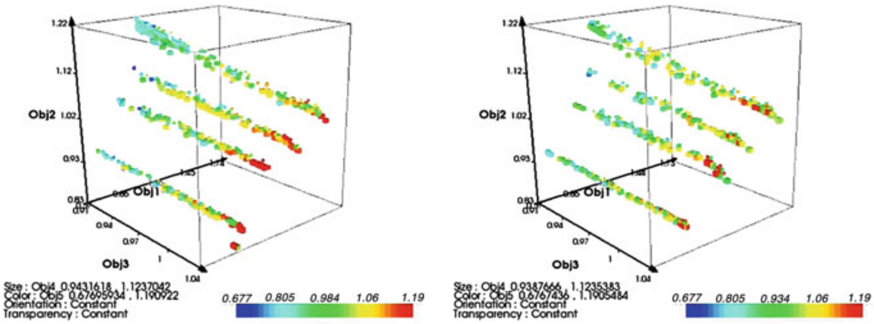


Fig. 7.7 Example of Trial 4 results: Pareto front generated by User 1 (*left*) and User 2 (*right*)

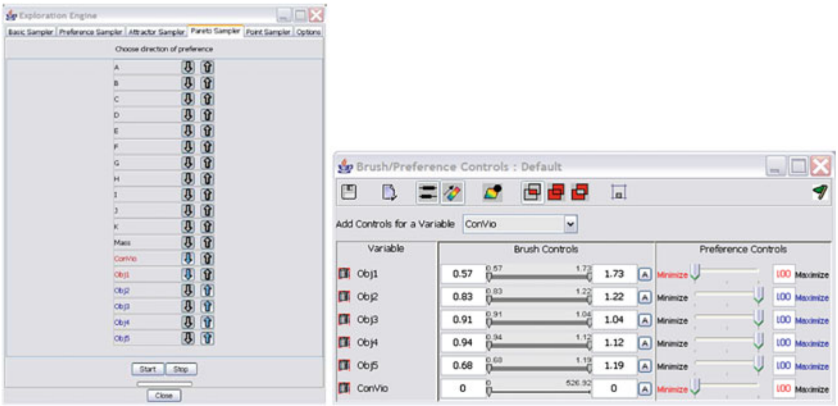


Fig. 7.8 Modifications to settings in ATSV for Trials 5–8: preferences for Pareto sampler (*left*) and brush/preference control settings (*right*)

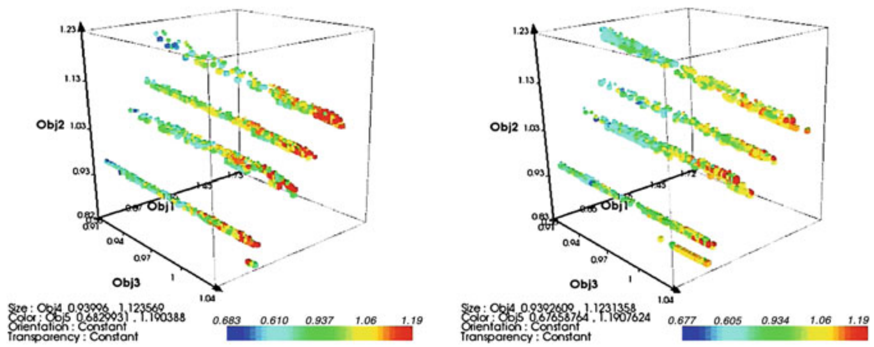


Fig. 7.9 Example of Trial 6 results: Pareto fronts generated by User 1 (*left*) and User 2 (*right*)

- Trial 4: Similar to Trial 2 with options changed to allow for more attractors.
- Trial 5: Tried to advance the Pareto front beyond what is visible.
- Trial 6: Tried to fill in the Pareto front with 3 objective interactions.
- Trial 7: Allow Pareto and Preference-based samplers to alternate and move through feasible space before restarting.
- Trial 8: Similar to Trial 7 with a different selection strategy.

These eight trails are based primarily on what felt "natural" to the users and represent only a fraction of the possible combinations.

7.4.3 Description of Reference Data Set

For comparison purposes, the reference (or "best known") Pareto front comes from an exhaustive multi-objective GA (MOGA) search that was performed previously on the same test problem by its originators (Ferguson et al. 2005a). In order to ensure that the Pareto front generated by the exhaustive MOGA contained no large holes or gaps (i.e., covered the entire objective space), a Gap Analyzer was developed that would direct the MOGA to find designs in those areas if such a region was found (Ferguson et al. 2005b). The exhaustive MOGA used approximately 80,000 function evaluations to create this reference Pareto front (44,769 points in the final population, 5,561 are Pareto-optimal over the continuous objective function space). Even with the Gap Analyzer, the MOGA ran "blindly", requiring no human intervention while searching the trade space; hence, it provides a suitable benchmark for this study.

7.4.4 Description of Performance Metrics

To quantify the performance and compare the results of genetic algorithms, a variety of performance metrics have been developed (Wu and Azarm 2001). Okabe et al. (2003) states that these metrics should be used to assess (1) the number of Pareto-optimal solutions in the set, (2) the closeness of the solutions to the theoretical Pareto-front, and (3) the distribution and spread of the solutions. Zitzler (1999) proposed a hyper-volume metric that evaluates the size of the dominated space. Tang et al. (2005) have developed and refined performance metrics to evaluate two Pareto fronts in a 5D trade space. In particular, their ε-performance has been used to assess the relative computational efficiency, accuracy, and ease-of-use, allowing simultaneous assessment of all three solution aspects advocated by Okabe et al.

The ε-performance metric developed by Kollat and Reed (2005a, b) was selected as the basis for comparison because it assesses the proportion of solutions found within a user-specified level of precision relative to the "true" Pareto front, or best available reference set. In other words, the user can specify a precision level for each objective to tailor it to a given application. The solutions are then evaluated with respect to the reference set based on this user specified precision. The proportion of reference set solutions found by the GA within this level of precision is reported as ε-performance. Since the solutions are evaluated with respect to a best known reference Pareto set, it is possible that the solutions may at times dominate reference set solutions. To account for this, ε-performance is reported in this study as the proportion of reference set solutions that are dominated, or found within the user-specified ε precision. These metrics allow for numerical comparisons between the solutions generated using the different combinations of visual steering commands within ATSV and the reference Pareto front obtained from the exhaustive MOGA that used 80,000 function evaluations.

7.5 Analysis and Discussion of Results

Figure 7.10 provides a visual comparison of the resulting Pareto fronts from individual trials along with the reference set. While it is difficult to make comparisons in 5D, we can identify from these figures the trials that did well and those that did not. For instance, Trial 4 by User 2 (see Fig. 7.10b) is much sparser, especially when compared to the reference set (see Fig. 7.10e).

Figure 7.10f provides a composite of all eight trials where the reference set from the exhaustive MOGA is shown in blue, solutions from Trials 1–4 and Trials 5–8 that are the same as the MOGA solutions are shown in green and red, respectively. As expected, the MOGA solutions dominate the majority of the solutions obtained from either sets of trials; however, it is promising to see that some solutions remain given that the trials used about 5,000 and 10,000 function evaluations compared to MOGA's 80,000.

Before comparing the sets of solutions quantitatively using the ε-performance metric, we need to determine a suitable value for epsilon. After confirming that all input and output variables were normalized by the same ranges and scaled against the same baseline values, we computed the differences between the objectives of every pair of designs in the reference set from MOGA. We found that the smallest difference between any two designs was so close to zero that any reasonable value of epsilon could be selected. While choosing an epsilon value that was too large would reduce each set to the point that comparison would be meaningless, choosing an epsilon value that was too small would make it almost impossible to find designs within one epsilon of each other in each objective given that it is a 5D space. Therefore, after performing a sensitivity study of epsilon values between 0.001 and 0.1, a value of 0.01 was selected for each objective and used for this analysis.

Table 7.2 shows the results of each trial using the ε-performance metric for both users (v1 and v2). As discussed in Sect. 7.4.4, the reported results are obtained by comparing each user's resulting Pareto set from each trial to the reference set

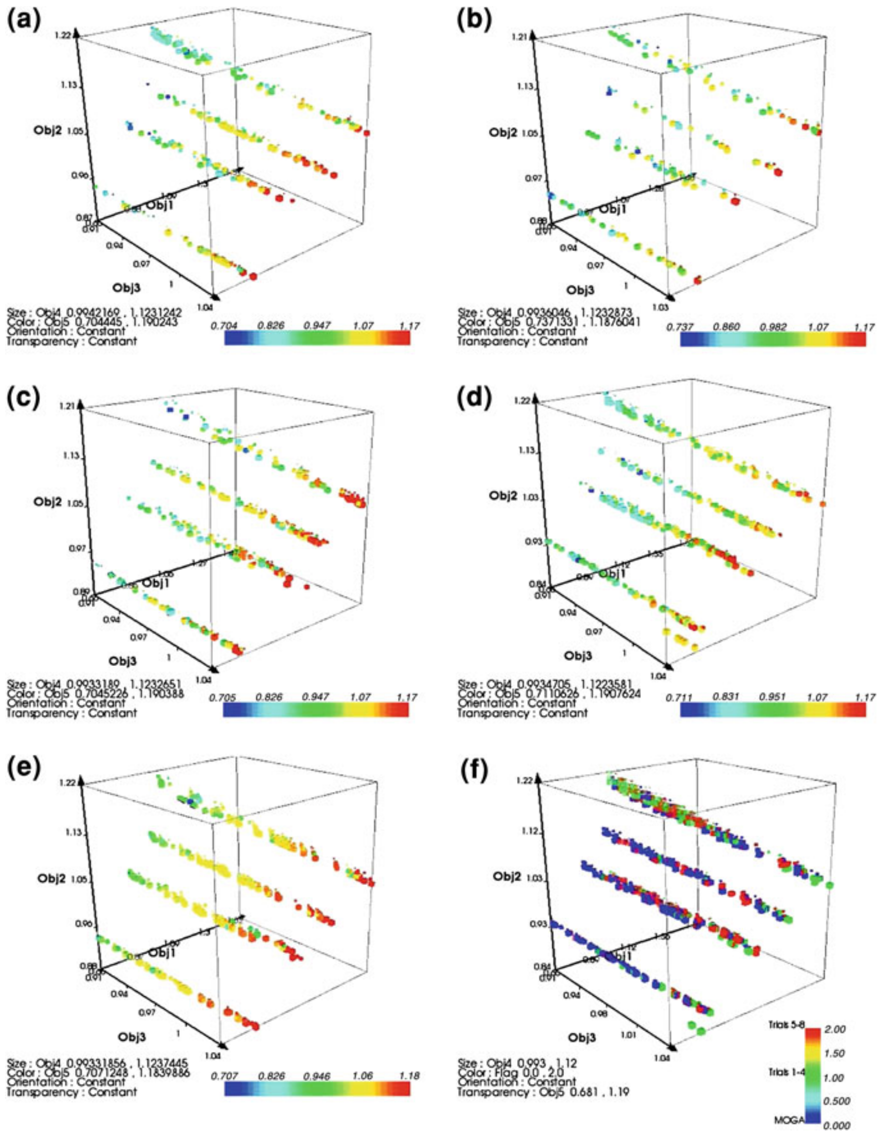


Fig. 7.10 Example of visual comparison of resulting Pareto fronts for Trial 4 (**a**, **b**), Trial 6 (**c**, **d**), the reference data set (**e**), and composite solutions color-coded by trial (**f**). **a** Trial 4 User 1 Pareto front. **b** Trial 4 User 2 Pareto front. **c** Trial 6 User 1 Pareto front. **d** Trial 6 User 2 Pareto front. **e** Reference Pareto front. **f** Pareto solutions color-coded by trial

obtained from the exhaustive MOGA, which is the best approximation of the "true" Pareto front that we can obtain. We see that when using only 5,000 points, both users are able to obtain 9.3–13.9% of the reference set; when allocated 10,000

Table 7.2 Summary of results based on the ε-performance metric for Trials 1–4 and Trials 5–8 for User 1 (v1) and User 2 (v2)

Trials 1–4 (∼5,000 points)									
Trial no.	1 v1	1 v2	2 v1	2 v2	3 v1	3 v2	4 v1	4 v2	Avg
Found (%)	0.00	0.61	0.40	0.20	0.40	1.01	0.20	0.61	0.43
Dominating (%)	12.90	10.69	9.68	13.71	11.69	12.90	13.71	8.67	11.74
Total	12.90	11.29	10.08	13.91	12.10	13.91	13.91	9.27	12.17
Trials 5–8 (∼10,000 points)									
Trial no.	5 v1	5 v2	6 v1	6 v2	7 v1	7 v2	8 v1	8 v2	Avg
Found (%)	0.40	0.00	0.61	0.20	0.40	0.61	0.20	1.21	0.45
Dominating (%)	14.31	12.90	21.37	22.18	17.34	19.36	17.94	18.15	17.94
Total	14.72	12.90	21.98	22.38	17.74	19.97	18.15	19.36	18.40

points, both users are able to increase this range to 12.9–22.4%. Thus, users are able to obtain, on average, 12 and 18% of the solutions on the Pareto front by using 1/16th and 1/8th, respectively, of the number of function evaluations used by the exhaustive MOGA.

Table 7.2 also shows that the 10,000 function evaluation trials perform better than the 5,000 function evaluation trials as one would expect, with Trial 6 performing the best. Not surprisingly, the percentage of designs found within epsilon of the reference solutions is very low for every trial. This is likely a result of the objective space being 5D, which makes it very difficult to find two designs that fall within 0.01 of each other in all five objectives. An unexpected result, however, is how high the percentage of designs dominating the reference is for each trial. This indicates that the reference set generated by the exhaustive MOGA is likely not the "true" Pareto set, but rather itself an approximation of the "true" set. The large percentage of dominating designs also illustrates the power of putting the human "in the loop" during optimization—even with only 5000 or 10,000 function evaluations, the user is already able to identify points that the MOGA did not find after executing its 80,000 function evaluations. This also shows that a user-guided trial in ATSV could possibly have a better chance of obtaining the "true" Pareto set than the MOGA running blindly.

To gain more insight into the performance of each trial as well as the evolution of solutions toward the Pareto front, we plot the ε-performance metric at a series of intervals leading up to the allocated number of function evaluations. In particular, Fig. 7.11 shows the performance of each user (v1 and v2) in Trials 1–4 at 500, 1,000, 2,000, 3,000, 4,000, and 5,000 function evaluations; Fig. 7.12 shows a similar progression for each user for Trials 5–8 at 500, 1,000, 2,500, 5,000, 7,500, and 10,000 function evaluations. In both figures, solutions from the exhaustive MOGA are also plotted based on its convergence history; so, for example, the ε-performance metric value plotted at the 500 function evaluation point indicates how well the MOGA has found the Pareto front by the time it has executed 500 of its 80,000 function evaluations.

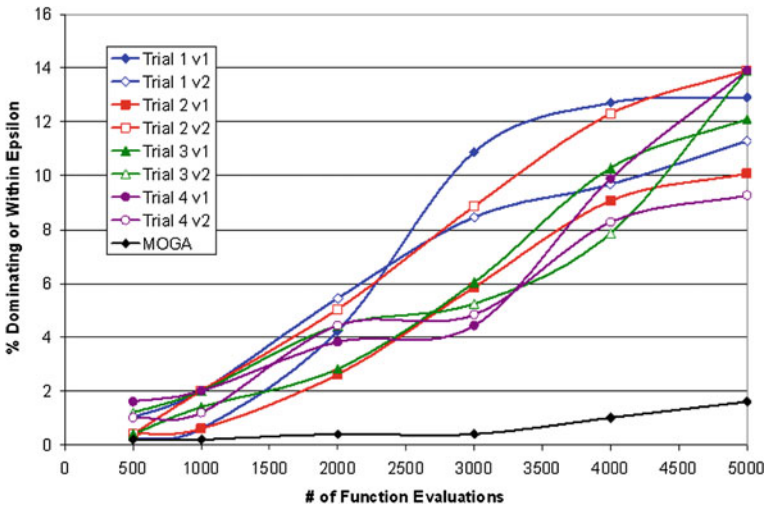


Fig. 7.11 Evolution of Pareto fronts in Trials 1–4 along with exhaustive MOGA search as measured by the ε-performance metric

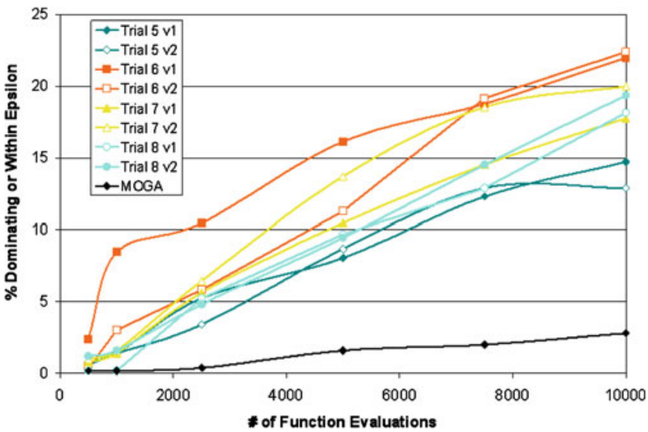


Fig. 7.12 Evolution of Pareto fronts in Trials 5–8 along with exhaustive MOGA search as measured by the ε-performance metric

While the results from Table 7.2 may not have been too convincing, Figs. 7.11 and 7.12 clearly illustrates the benefit of having the user "in-the-loop" during the optimization process. In all trials, both users have substantially out-performed the exhaustive MOGA in terms of the percentage of solutions found on the Pareto front (i.e., the reference set) for a given number of function evaluations. In Fig. 7.11, the MOGA has obtained fewer than 2% of the Pareto front in its first

5000 function evaluations compared to 9.3–13.9% in Trials 1–4. Likewise, even when the number of function evaluations has doubled to 10,000, the MOGA has still found fewer than 3% of the Pareto front compared to the 12.9–22.4% obtained in Trials 5–8 as indicated in Fig. 7.12. In both cases, this represents a 4–7× increase in the number of Pareto solutions obtained for a given number of function evaluations when the human is allowed to visualize and "steer" the optimization process. This increase is consistent, regardless of the combination of visual steering commands used or the designer implementing them. We note, however, that the total time for each user is not included in the study, i.e., the time each user spent visualizing and interpreting the results before determining the next sampler to use or direction in which to steer the search. Future studies should compare the total time taken by the users against the total time taken by the MOGA (and subsequent runs of the Gap Analyzer) to ensure a more accurate assessment of the benefits of including the human "in the loop" during optimization.

7.6 Conclusions and Future Work

Trade space exploration is a promising alternative decision-making paradigm that provides a visual and more intuitive means for formulating, adjusting, and ultimately solving design optimization problems. The results of this study indicate that the visual steering commands—regardless of the combination in which they are invoked—can provide a 4–7× increase in the number of Pareto solutions obtained for a given number of function evaluations when the human is "in-the-loop" during the optimization process. This user-guided search is also effective in identifying new Pareto points in as few as 5000 and 10,000 function evaluations that the multi-objective genetic algorithm had not found after 80,000 function evaluations. As such, this study provides empirical evidence of the benefits that interactive visualization-based strategies can provide in support of engineering design optimization and decision-making.

There are several possible extensions of this work. Additional metrics should be considered for comparing the solutions in the resulting Pareto fronts in terms of both the design variables (inputs) as well as the objective function values (outputs). A multi-metric strategy would be useful in not only assessing the goodness of the Pareto fronts more thoroughly but also providing guidance to users if they were computed in real-time during the trade space exploration process. Moreover, new metrics for measuring the insight gained during visual exploration are needed (North 2006) and would be very useful to convey the benefits of using trade space exploration. If we can assess the insight gained through visualization, it would be useful to repeat the study with domain experts (e.g., vehicle designers) who may employ specific search strategies or heuristics to solve the problem and see how their results compare. Based on the results in Table 7.2, we also need a more extensive reference set given the high dimensionality of the trade space being explored. Having a more complete reference set would prevent any individual trial from being able to dominate the reference set (or any portion thereof); trials would only be able to find points in the reference set within epsilon. Finally, the study should also be repeated with test problems of different sizes and complexity as well as with users with different levels of experience to demonstrate how widely applicable—and beneficial—the trade space exploration process is.

Acknowledgments We thank Dr. Scott Ferguson for his assistance with the vehicle model and the results from the exhaustive MOGA. This work has been supported by the National Science Foundation under Grant No. CMMI-0620948. Any opinions, findings, and conclusions or recommendations presented in this chapter are those of the authors and do not necessarily reflect the views of the National Science Foundation.

7.7 Appendix

7.7.1 Exploration Strategy for Trial 1 (Total Points = 5,025)

- Basic Sampler: 100 runs.
- Brush objectives 1–5: minimize Obj1 (−100), maximize Objs2–5 (100).
- Point attractors: 10 possible pair-wise point attractors for Objs1–5 set at the current limits of the scatter plot window (on Objs [1 & 2], [3 & 4], [5 & 1], [2 & 3], [4 & 5], [1 & 3], [2 & 4], [3 & 5], [4 & 1], [5 & 2]).
- Pareto Sampler.

7.7.2 Exploration Strategy for Trial 2 (Total Points = 5,075)

- Basic sampler: 500 runs.
- Brush Objs1–5: minimize Obj1 (−100), maximize Objs2–5 (100).
- Pareto Sampler.
- Line attractors (1D point attractor): one for each Obj1–5 set at the current limit of the scatter plot window (minimum of window for Obj1 and maximum of window for Objs2–5).
- Preference Sampler.
- Point attractors: set at current limits of the scatter plot window (on Objs [2 & 5], [2 & 4]).
- Point attractors: set at the current limits of the scatter plot window, generation size changed to 15 (on Objs [3 & 2], [3 & 4], [1 & 5], [2 & 5]).
- Point attractor: set at the current limits of the scatter plot window (on Objs [3 & 5]).

7.7.3 Exploration Strategy for Trial 3 (Total Points = 5525)

- Basic Sampler: 500 runs.
- Brush Objs1–5: minimize Obj 1 (−100), maximize Objs2–5 (100).
- Point attractors: set at the current limits of the glyph plot (on Objs [1, 2, & 3], [1, 2, & 4], [1, 2, & 5], [1, 3, & 4], [1, 3, & 5], [1, 4, & 5], [2, 3, & 4], [2, 3, & 5], [2, 4, & 5], [3, 4, & 5]).
- Pareto Sampler.

7.7.4 Exploration Strategy for Trial 4 (Total Points = 5375)

- Basic Sampler: 100 runs.
- Brush Objs1–5: minimize Obj1 (−100), maximize Objs2–5 (100).
- Line attractors: set at the current limits of the scatter plot window (on Objs1–5).
- Pareto Sampler.
- Point attractors: set at the current limits of the scatter plot window, generation size changed to 15 and population limit changed to 250 (on Objs [1 & 2], [1 & 3], [1 & 4], [1 & 5], [2 & 3], [2 & 4], [2 & 5], [3 & 4], [3 & 5], [4 & 5]).
- Line attractor (1D point attractor): set Obj3 at current limit of the scatter plot window.
- Point attractors: set at current limits of the scatter plot window (on Objs [3 & 4], [4 & 5]).

7.7.5 Exploration Strategy for Trial 5 (Total points: 10,375)

- Basic Sampler: 100 runs.
- Brush Objs1–5: minimize Obj1 (−100), maximize Objs2–5 (100).
- Point attractors: set at the current limits of the scatter plot window ±5% for minimizing or maximizing, respectively (on Objs [1 & 2], [2 & 3], [3 & 4], [4 & 5], [5 & 1], [1 & 3], [3 & 5], [5 & 2], [2 & 4], [4 & 1]).
- Preference Sampler.
- Point attractors: these specific values were used to fill in the Pareto front ([Obj1 = 0.9, Obj2 = 1.102], [Obj1 = 0.645, Obj2 = 0.872], [Obj2 = 1.144, Obj3 = 0.988]).
- Line attractors (1D point attractors): these specific values were used to fill in the Pareto front ([Obj4 = 1.124], [Obj5 = 1.191]).
- Brush (preference): minimize ConVio (−100).
- Preference Sampler.

- Pareto Sampler.
- Line attractors (1D point attractors): one for each Obj1–5 set at the feasible limit of the objective in the scatter window (minimum for Obj1 and maximum for Objs2–5).
- Pareto Sampler.

7.7.6 Exploration Strategy for Trial 6 (Total points: 10,375)

- Basic Sampler: 250 runs.
- Brush Objs1–5 and ConVio: minimize Obj1 and ConVio (−100), maximize Objs2–5 (100).
- Preference Sampler: generation size changed to 50 and population limit changed to 1,000.
- Pareto Sampler: generation size changed to 50 and population limit changed to 1,000.
- Point attractors: set at the current limits of the scatter plot window (on [ConVio & Obj1], [ConVio & Obj2], [ConVio & Obj3], [ConVio & Obj4], [ConVio & Obj5]).
- Pareto Sampler Generation size changed to 50 and population limit changed to 1,000.
- Point attractors: these specific values were used to fill in the Pareto front ([ConVio = 0, Obj1 = 1.043, Obj2 = 1.2], [ConVio = 0, Obj1 = 0.755, Obj3 = 1.026], [ConVio = 0, Obj1 = 0.911, Obj4 = 1.121], [ConVio = 0, Obj1 = 0.729, Obj2 = 1.153], [ConVio = 0, Obj2 = 1.126, Obj3 = 0.993], [ConVio = 0, Obj2 = 1.186, Obj4 = 1.099], [ConVio = 0, Obj2 = 1.154, Obj5 = 1.052], [ConVio = 0, Obj3 = 1.018, Obj4 = 1.123], [ConVio = 0, Obj3 = 1.003, Obj5 = 1.137], [ConVio = 0, Obj4 = 1.121, Obj5 = 1.105], [ConVio = 0, Obj3 = 0.923, Obj5 = 0.993], [ConVio = 0, Obj2 = 1.207, Obj5 = 0.853]).
- Preference Sampler.
- Pareto Sampler.
- Point attractors: use these specific values to fill in the Pareto front ([Obj1 = 0.802, Obj2 = 0.851, Obj3 = 1.007], [Obj3 = 1.003, Obj2 = 0.854], [Obj1 = 1.073, Obj2 = 1.19], [Obj4 = 0.995, Obj5 = 0.824], [Obj3 = 0.955, Obj4 = 1.119]).
- Pareto Sampler: population limit changed to 250.

7.7.7 Exploration Strategy for Trial 7 (Total points = 10,125)

- Basic Sampler: 25 runs.
- Brush Objs 1–5 and ConVio: minimize Obj1 and ConVio (−100), maximize Objs2–5 (100).
- Pareto Sampler: generation size changed to 50 and population limit changed to 1,000.

- Preference Sampler: generation size changed to 50 and population limit changed to 1,000.
- Repeat Pareto and Preference Samplers in above order with the same settings four more times.
- Pareto Sampler: generation size changed to 50 and population limit changed to 1,000.

7.7.8 Exploration Strategy for Trial 8 (Total points = 10,275)

- Basic Sampler: 25 runs.
- Brush Objs 1–5 and ConVio: minimize Obj1 and ConVio (−100), maximize Objs2–5 (100).
- Pareto Sampler: generation size changed to 50, population limit changed to 1,000, and selection strategy changed to Rand1Bin.
- Preference Sampler: generation size changed to 50, population limit changed to 1,000, and selection strategy changed to Rand1Bin.
- Repeat Pareto and Preference Samplers in above order with the same settings four more times.

References

Balling R (1999) Design by shopping: a new paradigm? In: Proceedings of the third world congress of structural and multidisciplinary optimization (WCSMO-3), University at Buffalo, Buffalo, New York, pp 295–297

Becker RA, Cleveland WS (1987) Brushing scatterplots. Technometrics 29(1):127–142

Buja A, McDonald JA, Michalak J, Stuetzle W (1991) Interactive data visualization using focusing and linking. In: Proceedings of the IEEE Conference on Visualization ’91, IEEE Computer Society Press, San Diego, CA, pp 156–163

Donndelinger J, Ferguson S, Lewis K (2006) Exploring mass trade-offs in preliminary vehicle design using pareto sets. 11th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Portsmouth, VA, AIAA-2006-7056

Dym CL, Wood WH, Scott MJ (2006) On the legitimacy of pairwise comparisons. Decision Making in Engineering Design, ASME, 135–143

Ferguson S, Gurnani A, Donndelinger J, Lewis, K (2005a) A study of convergence and mapping in multiobjective optimization problems. ASME design engineering technical conferences & Computers and information in engineering conference, Long Beach, CA, ASME, Paper No. DETC2005/CIE-84852

Ferguson S, Gurnani A, Donndelinger J, Lewis K (2005b) An approach to feasibility assessment in preliminary design. ASME design engineering technical conferences—design automation conference, Long Beach, CA, ASME, Paper No. DETC2005/CIE-84853

Kesavadas T, Sudhir A (2000) Computational steering in simulation of manufacturing systems. In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation. IEEE, San Francisco, CA, pp 2654–2658

Kollat JB, Reed PM (2005a) Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. Adv Water Resour 29(6):792–807

Kollat JB, Reed PM (2005b) The value of online adaptive search: aperformance comparison of NSGAII, e-NSGAII, and eMOEA. Lecture Notes in Computer Science, Springer, Berlin, p 3410

Madar J, Abonyi J, Szeifert F (2005) Interactive particle swarm optimization. International Conference on Intelligent Systems Design and Applications, IEEE, Wroclaw, Poland, pp 314–319

Messac A, Chen X (2000) Visualizing the optimization process in real-time using physical programming. Eng Optim 32(6):721–747

Michalek J, Papalambros P (2002) Interactive design optimization of architectural layouts. Eng Optim 34(5):485–501

Miettinen K, Makela MM (2006) Synchronous approach in interactive multiobjective optimization. Eur J Oper Res 170:909–922

North C (2006) Toward measuring visualization insight. IEEE Comp Graph Appl 26(3):6–9

Okabe T, Jin Y, Sendhoff B (2003) A critical survey of performance indices for multi-objective optimisation. Evol Comput 2:878–885

Price K, Storn R, Lampinen J (2005) Differential evolution: a practical approach to global optimization. Springer, Berlin

Robic T, Filipic B (2005) DEMO: Differential evolution for multiobjective optimization. Third international conference on evolutionary multi-criterion optimization, Guanajuato, Mexico, Springer, pp 520–533

Scott SD, Lesh N, Klau GW (2002) Investigating Human–Computer optimization. CHI'02, Minneapolis MN 4(1):155–162

Shanteau J (1992) Competence in experts: the role of task characteristics. Organ Behavior and Human Decis 53(2):252–266

Simpson TW, Donndelinger JA, Yukish M, Stump G (2007) Visual steering commands for trade space exploration: user-guided sampling with example. In: Proceedings of the ASME 2007 international design engineering technical conferences & computers and information in engineering conference, Las Vegas, NV, DETC2007/DAC-34684

Stump G, Yukish M, Simpson TW (2004a) The advanced trade space visualizer: an engineering decision-making tool. 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY, AIAA-2004-4568

Stump G, Yukish M, Simpson TW, Harris EN, O'Hara JJ (2004b) Trade space exploration of satellite datasets using a design by shopping paradigm. IEEE Aerospace Conference, IEEE, Big Sky, MT

Tang Y, Reed PM, Wagener T (2005) How effective and efficient are mulitobjective evolutionary algorithms at hydrologic model calibration. Hydrol Earth Syst Sci Discuss 2:2465–2520

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. ASME J Mech Des 129(4):370–380

Wilson TD, Schooler JW (1991) Thinking too much: introspection can reduce the quality of preferences and decisions. J Pers Soc Psychol 60(2):181–192

Winer EH, Bloebaum CL (2001) Visual design steering for optimization solution improvement. Struct Optim 22(3):219–229

Winer EH, Bloebaum CL (2002) Development of visual design steering as an aid in large-scale multidisciplinary design optimization Part I: method development. Struct Multidiscipl Optim 23(6):412–424

Wright H, Brodlie K, David T (2000) Navigating high-dimensional spaces to support design steering. Proceedings of IEEE visualization 2000, IEEE Computer Society Press, Salt Lake City, UT, pp 291–296

Wu J, Azarm S (2001) Metrics for quality assessment of a multiobjective design optimization solution set. ASME J Mech Des 123(1):18–25

Zitzler E (1999) Evolutionary algorithms for multiobjective: methods and applications. Swiss Federal Institute of Technology, Zurich, Switzerland

Chapter 8
Analyzing Global Epidemiology of Diseases Using Human-in-the-Loop Bio-Simulations

Dhananjai M. Rao, Alexander Chernyakhovsky and Victoria Rao

Abstract Humanity is facing an increasing number of highly virulent and communicable diseases such as influenza. Combating such global diseases requires in-depth knowledge of their epidemiology. The only practical method for discovering global epidemiological knowledge and identifying prophylactic strategies is simulation. However, several interrelated factors, including increasing model complexity, stochastic nature of diseases, and short analysis timeframes render exhaustive analysis an infeasible task. An effective approach to alleviate the aforementioned issues and enable efficient epidemiological analysis is to manually steer bio-simulations to scenarios of interest. Selective steering preserves causality, inter-dependencies, and stochastic characteristics in the model better than "seeding", i.e., manually setting simulation state. Accordingly, we have developed a novel Eco-modeling and bio-simulation environment called SEARUMS. The bio-simulation infrastructure of SEARUMS permits a human-in-the-loop to steer the simulation to scenarios of interest so that epidemics can be effectively modeled and analyzed. This article discusses mathematical principles underlying SEARUMS along with its software architecture and design. In addition, the article also presents the bio-simulations and multi-faceted case studies conducted using SEARUMS to elucidate its ability to forecast timelines, epicenters, and socio-economic impacts of epidemics. Currently, the primary emphasis of SEARUMS is to ease global epidemiological analysis of avian influenza. However, the methodology is sufficiently generic and it can be adapted for other epidemiological analysis required to effectively combat various diseases.

D. M. Rao (✉)
CSE Department, Miami University, Oxford, OH 45056-1601, USA
e-mail: raodm@muohio.edu

A. Chernyakhovsky
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
e-mail: achernya@mit.edu

V. Rao
Cybernetic Evolution, Mason, OH 45040, USA

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_8,

8.1 Introduction

Communicable and vector-borne diseases, primarily induced by viral and bacterial infections, are the most common forms of ailments amongst humans and domesticated farm animals (Flint et al. 2004). Despite numerous advances in epidemiology and medicine many of these diseases, particularly zoonotic diseases, continue to defy human efforts to treat and control them (Flint et al. 2004). The source of such resistance to treatment and control is twofold. First, the incessant evolutionary processes such as genetic reassortment, recombination, and mutation induce changes to organisms, thereby constantly morphing their antigenic characteristics (Russell et al. 2008). Second, the vectors of the diseases, such as birds and animals, often disperse the pathogens over wide areas (Hagemeijer and Mundkur 2006; Normile 2006). The scenario is further convoluted due to complicated symbiotic processes between the vectors and other organisms including humans. Rapid changes in phylogeny and bio-diversity pose significant challenges in vaccine manufacturing (WHO 2007a). In addition, socio-political restrictions and technological limitations further magnify the challenges involved in controlling intercontinental spread of infectious diseases (WHO 2007a).

The aforementioned epidemiological, antigenic, and socio-political traits of communicable diseases are salient characteristics of several recent global epidemics, such as: severe acute respiratory syndrome (SARS) (Hufnagel et al. 2004), avian influenza (H5N1) outbreaks (WHO 2006), and swine flu (H1N1) outbreak (Garten et al. 2009; Smith et al. 2009). Such emergent viral epidemics cause significant morbidity and mortality, both in humans and livestock (CDC 2006; WHO 2005). For example, human influenza-A viruses causes more than 500,000 moralities in humans annually (Russell et al. 2008). Its highly pathogenic avian homologe, namely H5N1, has caused economic devastation, particularly to poultry farming, in excess of 10 billion dollars (Kilpatrick et al. 2006).

The primary approach for combating emergent diseases, particularly viral diseases, is proactive, targeted antiviral prophylaxis (CDC 2006; WHO 2007b). Unfortunately, the constantly changing antigenic characteristics of viruses, particularly all influenza viruses, reduce efficacy of vaccinations (WHO 2006c). Moreover a myriad of technological issues pose serious hurdles to manufacturing and distribution of even small volumes of H5N1 vaccines (WHO 2006c). Furthermore, intercontinental migration of birds and international human travels further exacerbate effective containment and mitigation of epidemics.

8.1.1 Epidemiological Simulations: A Brief Overview

The technological, macro-economic, and socio-political issues surrounding emergent, viral and communicable diseases (discussed in Sect. 8.1) have led to a rapidly growing emphasis on analyzing the global epidemiology of diseases.

Epidemiology broadly covers the study of various facets of communicable diseases in a selected population, including: transmission characteristics of diseases, epidemic and endemic states, environmental factors impacting disease transmission, and impacts of containment and prophylactic interventions (Anderson and May 1992). Epidemiology is a multidisciplinary method involving biology, biostatistics, geographic information science, social science, and computer science (Anderson and May 1992; Daley and Gani 2001). It has steadily evolved from its inception in early nineteenth century and is now a mainstream methodology that forms the corner stone for public health measures and preventive medicine (Anderson and May 1992). Epidemiological analysis enables more accurate forecasting of epidemics and aids improving efficacy of geographically targeted antiviral prophylaxis (Epstein 2009; Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005). Nevertheless, epidemiology is an active area of research as it continues to evolve in conjunction with breakthroughs in other disciples, such as: microbiology, genetics, proteomics, and meta-genomics (Rao et al. 2007a).

A fundamental and widely used framework underlying epidemiological analysis is compartmentalized model of communicable diseases (Anderson and May 1992; Daley and Gani 2001). In a compartmentalized model, the subpopulation that are in the same epidemic state are partitioned into non-intersecting sets called compartments (Anderson and May 1992; Daley and Gani 2001). The temporal progress of an epidemic is modeled by transitioning a suitable fraction of the population from one compartment to another (Anderson and May 1992; Daley and Gani 2001). The transition functions are typically modeled as ordinary differential equations (ODEs) with time as the independent variable. Constants in the ODEs are determined based on characteristics of the disease being analyzed. A more detailed description of the compartmentalized models is presented in Sect. 8.2.1.

Analysis of compartmentalized epidemiological models is performed by simultaneously solving the ODEs at different time steps to obtain characteristics of the epidemic being studied (Anderson and May 1992; Daley and Gani 2001). One of the most powerful and widely used approaches for such an analysis is computer-based simulation. Simulations have gained significant importance epidemiology because they are the only practical approach to analyze large and complex epidemiological models (Epstein 2009; Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005; Rao and Chernyakhovsky 2008). Furthermore, the need to conduct a variety of multi-faceted analysis within short time frames necessitates the use of computer-based simulations. Moreover, simulation is a cost effective and non-destructive methodology from which results can be easily displayed in an intuitive form. More importantly, it enables effective explorations of policies and procedures associated with complicated control measures, such as: targeted layered containment [the main concept behind US government's containment strategies (Halloran et al. 2008)], quarantine, social distancing, school closing, and targeted antiviral prophylaxis (Epstein 2009; Ferguson et al. 2006; Longini et al. 2005; Rao et al. 2007a).

8.1.2 Motivation for Human-in-the-Loop Simulation

Currently, epidemiological models are used by many international agencies, including the World Health Organization (WHO) and the Centers for Disease Control (CDC), for large-scale, multi-faceted analysis required to propose and validate multi-national targeted layered containment policies as well as prophylactic measures (Epstein 2009; Ferguson et al. 2006; Longini et al. 2005; Rao et al. 2007a). Such epidemiological analysis is performed using detailed, stochastic models that involve complex, symbiotic interactions between the various entities involved in the simulation.

Typically, only a selected subset of scenarios is analyzed to evaluate effectiveness of a candidate set of containment strategies (Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005). Narrowing the subset of scenarios to be analyzed is critical because hundreds of thousands of simulation runs are needed even to analyze a specific subset of scenarios. The numerous runs are necessary to provide sufficiently accurate and statistically significant results using realistic, stochastic models (Epstein 2009; Ferguson et al. 2006; Longini et al. 2005).

One of the important prerequisites of simulating selected scenarios is to have the model in a state that accurately reflects the conditions associated with the scenario. One strategy is to seed the simulation, i.e., manually initialize the state to appropriately model the scenario being analyzed. However, such an approach is typically tedious, cumbersome, and error prone. Moreover, in stochastic models, accurately capturing temporal, causal relationships (Lamport 1978) between entities in the model can prove to be a greater challenge. These issues often lead to conspicuous inefficiencies when multiple scenarios have to be analyzed. An alternative approach is to commence simulations in a given, verified initial state and then steer the simulations to desired scenarios of interest. Strategies for steering simulations can be broadly classified into three main categories, namely: automatic, semi-automatic, and manual steering. Note that this taxonomy is based on the degree of human involvement in the steering process.

The complex, stochastic nature of global epidemiological models necessitates the use of manual steering. Manual steering of a simulation requires a human-in-the-loop (HITL) to intermittently modify the state of the simulation; thereby altering its trajectory to the desired scenario. Such HITL epidemiological bio-simulations enable efficient generation of various scenarios to be analyzed. In addition, this approach preserves the causal inter-dependencies in a stochastic model. Moreover, simulating from a validated initial state minimizes modeling errors and reduces continued verification efforts. Last but not the least, HITL steering enables some degree of adaptation and modeling of unplanned or unanticipated scenarios that could occur in emergent epidemics. Consequently, HITL-simulations hold significant potential to play a vital role in epidemiology and in establishing pertinent public policies.

8.1.3 SEARUMS: A Human-in-the-Loop Epidemiological Simulator

Realizing the advantages of simulation-based epidemiological analysis requires the use of an effective software environment for Eco-modeling, HITL bio-simulation, and analysis (Rao et al. 2007a). In addition, the software environment must facilitate rapid simulation to minimize analysis time frames. We also envision the software must be portable and accessible to enable its widespread use (Rao et al. 2007a). Accordingly, we have endeavored to design and develop an Eco-modeling and bio-simulation software environment called SEARUMS.

Currently, SEARUMS is geared for analyzing the global epidemiology of avian influenza, with migrating waterfowl as the primary vectors for intercontinental spread of the disease. However, the methodology is sufficiently generic and can be easily adapted for other diseases. Accordingly, this chapter elucidates the modeling, HITL-simulation, and software design principles underlying SEARUMS so that the concepts can be readily adapted and applied to other diseases as well. Furthermore, SEARUMS is envisioned to serve as a global, multi-disciplinary environment that seamlessly integrates knowledge from various fields so that epidemiologists, economists, and disease control centers can collaboratively use it to combat global epidemics.

8.1.4 Section Organization and Audience

The principles underlying the Eco-modeling and HITL bio-simulation infrastructure of SEARUMS span a broad range of disciplines, including: mathematical modeling, statistical analysis, computer science, and software engineering. Furthermore, the multi-disciplinary nature of SEARUMS garners interest from a broad range of audience, including: epidemiologists, economists, software architects, and disease control centers. Consequently, the various sections in this chapter are organized to emphasize a specific aspect of SEARUMS. The objective of this organization is to enable readers from a particular discipline to focus on pertinent topics. Accordingly, common concepts that are pervasive across the various sections are discussed as the background information in Sect. 8.2. This section also reviews some of the pertinent related research investigations. Section 8.3 discusses the software architecture and design principles underlying SEARUMS. The procedure used for modeling, incorporation of real-world statistical data, and analysis using SEARUMS is presented in Sect. 8.4. Results from some of the analysis conducted using SEARUMS are summarized in Sect. 8.5. Section 8.6 concludes the chapter while emphasizing the utility and broader applicability of the HITL-simulations infrastructure of SEARUMS to other diseases.

8.2 Background and Related Research

Simulation-based analysis of global epidemiology of diseases is a multi-disciplinary task that involves ecology, mathematics, statistics, computer science, and software engineering (Anderson and May 1992; Epstein 2009; Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005; Rao et al. 2009). Accordingly, this section covers relevant prerequisite information and terminology on the aforementioned topics along with brief surveys of closely related research investigations. Specifically, Sect. 8.2.1 covers the compartmental models used in epidemiology. Although the compartmental models can be applied to many diseases, in this article we focus on its use for analyzing global epidemiology of avian influenza. Consequently, the ecology of avian influenza is summarized in Sect. 8.2.2 to further motivate the Eco-modeling and bio-simulation environment constituting SEARUMS. Next Sect. 8.2.3 presents an overview of the Markov processes that are used to model the natural interactions occurring in the ecology. This section also illustrates the interactions between the compartmental models and Markov processes. Section 8.2.4 presents a brief survey of the concepts and software frameworks pertaining to HITL steering of simulators and software systems.

8.2.1 Compartmental Models in Epidemiology

The most widely used mathematical framework for epidemiological analysis are compartmentalized models (Anderson and May 1992; Daley and Gani 2001). In a compartmentalized model the population being analyzed is partitioned into a few non-intersecting subsets called compartments. Compartments are defined such that the sub-population within a compartment exhibits a vital disease characteristic (Anderson and May 1992; Daley and Gani 2001), such as:

- *Susceptible*: population lacks immunity
- *Exposed*: infected, but not yet infectious sub-population
- *Infected*: sub-population is actively spreading the disease, and
- *Recovered*: sub-population acquired immunity or died due to disease.

Additional compartments are used to appropriately model other epidemiological states of diseases depending on the analysis needs (Epstein 2009; Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005; Rao et al. 2007a).

The classical susceptible-exposed-infected-removed (SEIR) compartmentalized mathematical model shown in Fig. 8.1 is used to model the epidemiology of various diseases (Anderson and May 1992). The characteristics of the SEIR model are represented using the following system of differential equations:

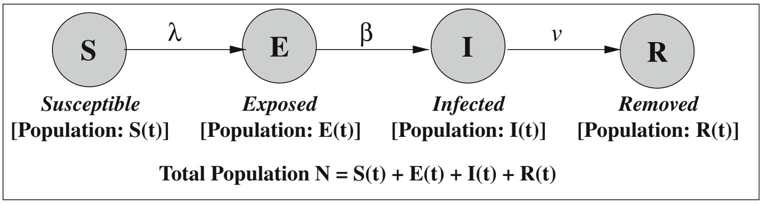


Fig. 8.1 A classical SEIR epidemiology model illustrating the four typical compartments used to describe the progression of a disease through a population (of N individuals)

$$\frac{dS}{dt} = \mu N - [\lambda + \mu]S(t)$$

$$\frac{dE}{dt} = \lambda S(t) - (\beta + \mu)E(t)$$

$$\frac{dI}{dt} = \beta E(t) - (v + \mu)I(t)$$

$$\frac{dR}{dt} = vI(t) - \mu R(t)$$

where, *S(t)*, *E(t)*, *I(t)*, and *R(t)* represent the number of susceptible, exposed, infected, and removed hosts at any given instant of time *t*. The temporal progress of an epidemic is modeled by transitioning a fraction of the population from one compartment to another (Anderson and May 1992; Daley and Gani 2001). The transition functions are typically modeled as ODEs with time as the independent variable.

Constants in the ODEs are determined based on characteristics of the disease being analyzed. Specifically, the parameters μ, λ, β, and v are:

1. μ: the per capita host birth/death rate,
2. λ: the force of infection
3. β: latency period
4. v: per capita recovery rate

Typically, the ODEs involve stochastic components to account for uncertainties introduced by various environmental or external factors (Anderson and May 1992). An important and distinguishing property of the compartmentalized models and the transition functions is that the total population (N in Fig. 8.1) being modeled is held a constant (Anderson and May 1992; Daley and Gani 2001).

Adaptations of the aforementioned classical SEIR model have been widely used to model the epidemiology of influenza (Epstein 2009; Ferguson et al. 2006; Halloran et al. 2008; Longini et al. 2005; Rao et al. 2007a). Investigations conducted by Longini et al. (2005) and Ferguson et al. (2006) focus on analyzing pandemic mode of H5N1 in Thailand. In pandemic mode rapid and sustained human-to-human transmission is assumed. Since human-to-human transmission is assumed, these investigations use a highly detailed spatially explicit model based

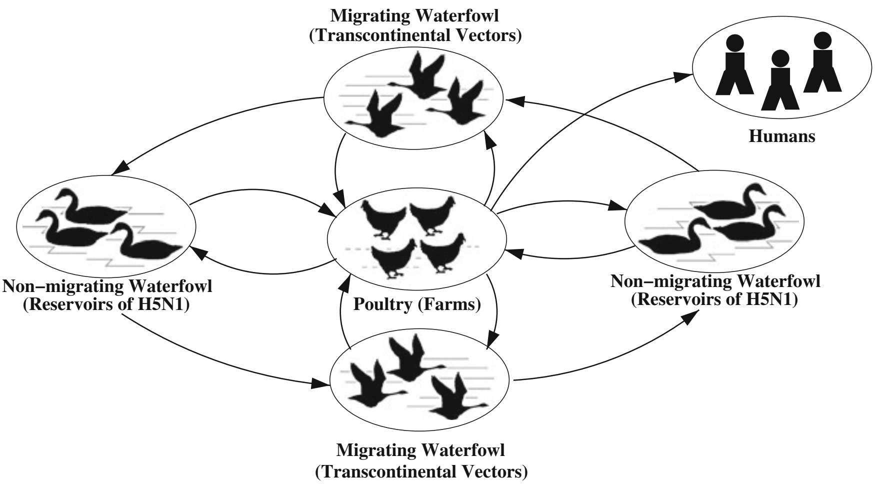


Fig. 8.2 The figure illustrates the salient pathways for transmission of H5N1 to various bird species and humans. The *arrows* in the diagram show the various directions of infection that occur in the ecology of avian influenza

on SEIR concepts. The modeling approach used by Halloran et al. (2008) is similar to those proposed by Ferguson et al. (2006) and Longini et al. (2005). Moreover, these three investigations are based on the premise that H5N1 has already mutated to a pandemic form and epidemics are being caused primarily due to human-to-human transmission.

In contrast, our investigations assume and reflect the current, real-world situation; i.e., H5N1 is yet to mutate into its pandemic state and human-to-human transmission is unsustained. Furthermore, our research aims to utilize waterfowl migration data to forecast zoonotic epicenters and timelines of potential epidemics. In addition, we also emphasize the role of poultry (extensible to swine) as intermediate hosts. These aspects notably distinguish our efforts from the aforementioned recent related investigations.

8.2.2 Avian Influenza

Avian influenza is a viral disease caused by H5N1, a highly virulent strain of the influenza-A virus, that has the potential to cause a global pandemic (CDC 2006; WHO 2006c). The ecological interactions contributing to the transcontinental spread of the disease is illustrated in Fig. 8.2. As shown in the figure, infected migrating waterfowl, in which the virus is endemic, are the primary vectors for causing intercontinental spread of the disease (Normile 2006).

The virus rapidly spreads from waterfowl to poultry and humans through contaminated water, feed, feces, and surfaces. Once infected, the disease has a devastating impact on poultry farms causing 100% mortality within 48 h (WHO 2007b). In humans, the virus causes disease with a high mortality rate of nearly 60% (CDC 2006; Normile 2006). Furthermore, it is known to induce primary viral pneumonia in the host (Normile 2006). Researchers believe that avian influenza has significant potential to become one of the deadliest pandemics in human history (CDC 2006; WHO 2006c). This inference has been drawn based on statistics from recent epizootic outbreaks and the highly pathogenic characteristics of H5N1. Moreover, manufacturing and distribution of vaccinations is facing multi-faceted challenges (WHO 2007a). The aforementioned issues make it imperative the epidemiology of avian influenza is thoroughly analyzed in order to empower various national and international organizations with the knowledge to strategically combat the disease (Anderson and May 1992; Epstein 2009; Ferguson et al. 2006; Longini et al. 2005; GLiPHA 2007; Rao et al. 2009).

8.2.3 Markov Processes

The compartmental epidemiological models focus purely on the temporal progression of the disease in a given entity, may it be waterfowl, human, or poultry. However, the compartmental SEIR models (see Sect. 8.2.1) do not embody the complete ecology, such as seasonal migration, occurring in nature. Consequently, such ecological processes need to be suitably modeled in conjunction with the epidemiological process. Accordingly, we envision using the concept of a Markov processes to model the overall ecological processes. Furthermore, our Eco-modeling approach suitably incorporates the SEIR models to provide a complete, holistic representation of the real-world ecology.

A Markov process is a mathematical formalism used to describe changes occurring to the state of a stochastic system in discrete time steps (Solow and Smith 2006; Winston 1994). A Markov process consists of a number of states (or values) through which the system may transition at any given time. Mathematically, a Markov process is defined as a sequence of time dependent random variables $X_0, X_1, X_2, \ldots$, where X_t is a random variable that describes the state of the process at discrete time t. The initial or starting state of the system is typically represented by X_0. Transitions from one state to another are governed by the following three laws: (1) a Markov process may be in only one given state at any instant of time; (2) transition from one state to another occurs instantaneously in discrete time steps; and (3) the next state to which the process transitions is purely determined by the current state of the system and not its past. In other words, the past, present, and future states of a Markov process are independent of each other (Winston 1994).

In our approach, the SEIR operations are repeatedly preformed in an appropriate state in the Markov process. It must be noted that the SEIR models involving

ODEs are inherently based on the notion of continuous simulation-time (Anderson and May 1992). However, the Markov processes operate in discrete simulation-time steps (Winston 1994). Consequently, our Markov process-based approach approximates the SEIR model. However, the discrete simulation-time steps are chosen to be sufficiently small to provide adequate accuracy without degrading performance (Rao et al. 2009).

Another important aspect of our Eco-modeling methodology is the aggregate representation of sub-populations. At a global scale, the varying populations of entities are typically modeled at a coarser granularity or aggregate entities (Booth 1997; GROMS 2006; Law et al. 2005; Rao et al. 2009). For example, certain species of waterfowl that live and migrate as a large flock in nature are typically modeled as a single entity. A large collection of birds, such as a poultry farms are represented as a single entity. In an analogous manner, humans living in geographic proximity to each other are modeled as a single entity.

The motivation for such a coarser grained, aggregate representation is two fold (Booth 1997; GROMS 2006; Law et al. 2005; Rao et al. 2009). First, such an approach is necessary to reduce the size and complexity of the model to more tractable scales. Aggregated models continue to provide the necessary fidelity as long as the aggregation is performed at a sufficiently fine granularity (Law et al. 2005). Second, various real-world statistical data on population dynamics, seasonal migratory behaviors, and disease progresses are determined through random sampling of a large subset of the population. Such data is more meaningful and relatively straightforward to apply at an aggregate level.

In our software system the three primary aggregate entities, namely: (1) flock of waterfowl; (2) poultry; and (3) group of humans have been modeled using the aforementioned Eco-modeling approach. The Markov processes have been suitably implemented using the modeling framework provided by SEARUMS. The Markov process for the waterfowl entity is shown in Fig. 8.3. Readers are referred to the literature for more details on the Markov processes for various entities and the associated mathematical equations (Rao et al. 2009).

8.2.4 Human-in-the-Loop Simulation and Steering

In conjunction with advancements in microprocessor technologies, computer-based simulations have become an important, indispensable, and multi-disciplinary methodology for study and analysis of complex systems (Railsback et al. 2006). Today, simulations are widely used in fundamental sciences, applied sciences, engineering, medicine, economics, and in the military (Tobias and Hofmann 2004). During its inception, simulations were run purely as an offline or batch processing task without requiring any interaction with a human operator. However, the growing demand and diversification in needs in various disciplines have led to the development of simulations which involve interactions with a human operator (Rao et al. 2007b). Such simulations, which permit or require a human to interact,

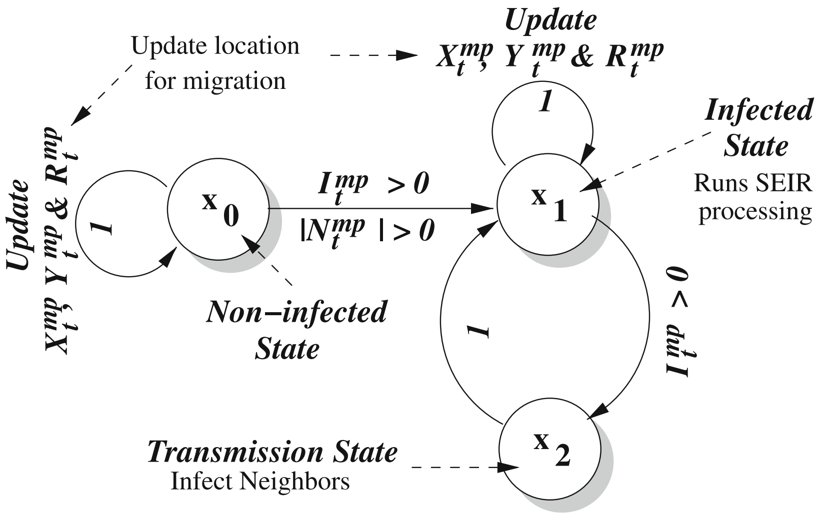


Fig. 8.3 Overview of Markov process for a waterfowl flock illustrating the states through which the process transitions to model the behavioral life cycle of waterfowl. The state of the process S_t^{mp} at discrete time t is represented by the four-tuple $S_t^{mp} = < X_t^{mp}, Y_t^{mp}, R_t^{mp}, I_t^{mp} >$, where X_t^{mp} is the longitude, Y_t^{mp} is the latitude, R_t^{mp} is the instantaneous radius of the flock, I_t^{mp} is the current measure of infection in the flock. The set N_t^{mp} represents the neighborhood at time *t*. Readers are referred to the literature (Rao et al. 2009) for details on the mathematical notations, spherical geometric equations, and state transitions

are classified as HITL-simulations. The process of impacting the state of the simulation to divert the simulation to another state is called simulation steering. Simulation steering is similar to program steering, which is a more generic concept (Gu et al. 1994).

HITL-simulations can be further classified into constructive HITL-simulations and virtual simulations. In a virtual simulation, simulation-time progressed as the same rate as the real, wall-clock time (Rao et al. 2007b). Furthermore, interaction between the human and the simulation occurs in a synchronous manner. Virtual simulations are primarily used for training, gaming, and education (Rao et al. 2007b). On the other hand, in a constructive simulation, simulation-time progresses at a different rate than real time. Human interactions with a constructive simulation are typically intermittent or on a "as needed" basis. The interactions are triggered based on an user gesture or when predefined events occur in the simulation.

Constructive simulations are used for simulating large and complex systems such as the ecological and epidemiological models. Several such general purpose simulators have been described in the literature (Gilbert and Bankes 2002; Railsback et al. 2006; WHO 2005). Some of the commonly used simulators include NetLogo, SWARM, SWARM-Java, Repast, and MASON. Railsback et al. (2006) highly recommended NetLogo for its ease-of-use. However, it uses a custom language for modeling and its source code is proprietary. On the other hand, the latter four platforms uses traditional programming languages and source codes are freely available (Railsback et al. 2006). These four platforms essentially provide a core framework for model development and a collection of library modules. The library modules are built using the core framework and can be readily reused for modeling. Moreover, the simulators provide some support for HITL interactions.

The aforementioned simulators are mostly general purpose simulators and are not specifically designed for epidemiology. However, our objective is to minimize learning curves for both developers and users, maximize portability, include intuitive interfaces for modeling, and seamlessly incorporate epidemiological analysis tools. Furthermore, the simulators had some disadvantages (Rao et al. 2009). Consequently, we endeavored to develop SEARUMS, a custom Eco-modeling and HITL bio-simulation environment.

8.3 SEARUMS

SEARUMS is an Eco-modeling and HITL bio-simulation environment. Currently, it is optimized to enable study and analysis of global epidemiology of avian influenza. However, the design of its modeling and simulation framework is sufficiently generic. Therefore, it can be adapted for epidemiological analysis of other diseases. SEARUMS has been developed in Java by utilizing many of the language's object oriented programming features (Bloch 2001). SEARUMS is designed to be a user friendly, integrated, graphical modeling, simulation, visualization, and analysis environment for conducting epidemiological analysis. These design goals have been achieved by composing the system using a collection of interdependent but loosely coupled modules as shown in Fig. 8.4.

Each module shown in Fig. 8.4 has a well-defined functionality that can be accessed and utilized via a set of application program interface (API) method calls. APIs of the modules are Java interface classes that are implemented by each module. Interactions between modules are achieved through interface classes to ensure loose coupling. This approach permits seamless "plug and play" of modules and the environment is composed by loading suitable modules dynamically on-demand via Java's reflection API (Bloch 2001). Such an implementation approach has been adopted to ease customization and extension of SEARUMS without requiring changes to its design or impacting existing modules.

The modules constituting SEARUMS are broadly classified as core modules and graphical user interface (GUI) modules. The core modules of SEARUMS are the agent repository, agent customizer, persistence module, HITL steering module, simulation module, and logging module. These modules provide the core M&S functionality of SEARUMS. The GUI facilitates interactions with the core modules via convenient and intuitive user interfaces. The GUI modules can be further categorized into the editor subsystem, the simulation controller, and the visualization and analysis subsystem. The GUI presented by these modules is shown in Fig. 8.5. SEARUMS uses the model-view-controller pattern to couple the core

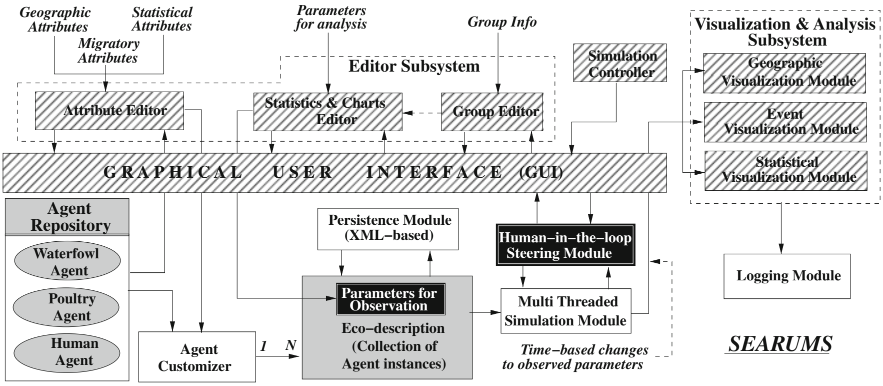


Fig. 8.4 Architectural overview of SEARUMS. GUI modules are highlighted with a striped background. SEARUMS can be downloaded from http://www.searums.org

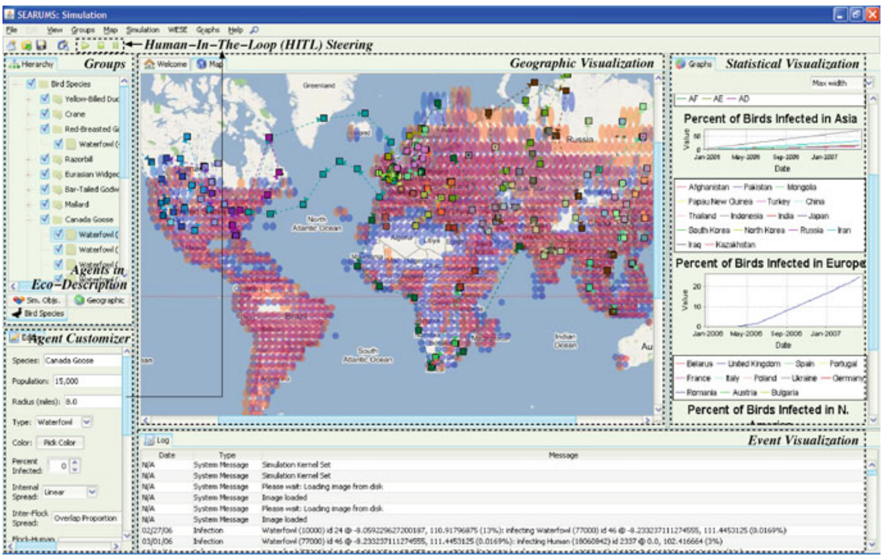


Fig. 8.5 The figure presents a screenshot of SEARUMS illustrating the graphical layout as seen by a user. The various modules constituting SEARUMS have been marked using *black dashed lines*. *Purple circles* indicate human groups and *orange circles* indicate poultry flocks. Variation in colors arises due to overlap of human and poultry flocks. *Colored squares* and corresponding *colored dashed* lines illustrate migration paths of waterfowl flocks

modules, the GUI models, and the Eco-description. The design permits the GUI modules to be easily replaced with a minimal command-line text interface for running SEARUMS in offline batch mode. The batch mode is useful for performing repeated runs or analyzing different scenarios on computational clusters.

The modules and subsystems constituting SEARUMS cooperatively operate on a shared, in-memory representation of the model called the Eco-description. The Eco-description is a centralized data structure that includes all the information necessary for modeling, simulation, and analysis. It is composed using a collection of Java classes and that provide efficient access to data and information required by the various modules. The primary information encapsulated by the Eco-description parameters for analysis relates to the smart agents (Hare and Deadman 2004) that constitute the model. As shown in Fig. 8.4, the agents are organized into an AGENT REPOSITORY to facilitate instantiation and use via Java reflection API.

Currently, SEARUMS includes the following three smart agents: WATERFOWL AGENT that represents a migrating waterfowl flock, POULTRY AGENT that models behavior of poultry flocks, and HUMAN AGENT that models humans. Each agent has its own behavior that reflects the characteristics of its real-world counterpart. The behaviors are customized to represent specific instances of an agent by specifying suitable values for the exposed attributes via the ATTRIBUTE EDITOR GUI module. The attributes of an agent include:

1. *Geographic attributes* that indicate the location (latitude and longitude) and logical association with countries and continents. In addition, each agent has a circle or influence that circumscribes its neighborhood.
2. *Migratory attributes* are specified only for agents whose location changes over the lifetime of the simulation. The migratory attributes are described as a sequence of migration points. Each migration point has geographical and chronological (arrival and departure dates) attributes associated with it. In SEARUMS, only one complete migration cycle needs to be specified. The software automatically reuses the information to simulate annual migratory cycles.
3. *Statistical attributes* for agent instances include their initial population, density and distribution, initial infection percentage, infection spread parameters, incubation periods, mortality rates, and population re-growth parameters.

The agents implement the conceptual, mathematical model of the system developed using Markov processes as described in Sect. 8.2.3. They are added to a model via suitable toolbar buttons or menu options provided by SEARUMS. Agent instances are created with default attributes from the agent repository by the AGENT CUSTOMIZER module using Java's reflection API. Once instantiated, the attributes for agents can be modified via the ATTRIBUTE EDITOR module. The agents are implemented as a family of Java classes by extending a common base class called AGENT. The AGENT class provides methods for interacting with the simulation kernel, inspecting the neighborhood, scheduling events, and interfacing with the GUI modules.

The agents in a model are logically organized into hierarchical sets called groups. SEARUMS permits multiple top-level groups with an arbitrary number of hierarchies, with one or more sub-groups at each hierarchical level. An agent can be a member of multiple groups. The groups serve several different purposes in

SEARUMS. A group can be used as a parameter for statistical analysis and for plotting charts. For example, a group called US can be created with 50 different sub-groups, one for each state, encompassing various agents. The main US group can be selected for plotting charts and SEARUMS automatically collates and plots data for each state. Note that even though graph plotting is restricted to one hierarchical level, statistics for plotting are collated in a recursive, depth-rst manner and includes data from all agents in underlying hierarchies. A modeler can use a combination of groups to perform multi-faceted analysis at different scales. In addition, groups can be included or excluded from simulations for analyzing different scenarios. The GUI modules utilize groups to provide control on visibility of agents to manage details displayed on the screen. The group editor module provides the user interface for managing group entries and hierarchies.

Once all the agent instances and groups have been established in a model, the parameters for observation are added to the Eco-description. These parameters are selected by the user via the Statistics and Charts editor from a list of options. The list includes the attributes of the agents and the groups in the Eco-description. Each parameter is configured to be sampled hourly, daily, or weekly in terms of simulation-time. Moreover, each parameter can be subjected to statistical operations, such as sum, mean, and median. SEARUMS can dynamically (i.e., during simulation) plot and save a variety of charts including: line graphs and pie charts. Multiple charts can be simultaneously used for analyzing a variety of data.

All of the aforementioned information is stored as an integral part of the Eco-description. The Eco-description can be saved for future reuse via the PERSISTENCE MODULE. The Eco-description is unmarshalled into a XML document that is compliant with a predefined XML schema. Serializing to a XML document provides a few advantages. First, it enables simple scripts to be developed that can modify specific values and perform multiple simulation runs in batch mode. Second, XML documents can be readily version controlled and archived using commonly available revision control systems such as CVS and subversion. Third, it eases documentation, validation, sharing, and reuse of valuable domain-specific statistical data collated by different researchers from diverse sources. Lastly, it is used to create checkpoints that reflect different scenarios. Such features play an important role in facilitating large-scale, collaborative epidemiological studies.

The SIMULATION MODULE performs the task of conducting a discrete event simulation (DES) using the Eco-description. This module utilizes a multi-threaded DES kernel that manages and schedules the discrete events generated by the agents. Multi-threading enables the DES kernel to exploit the compute power of multiprocessor or multi-core machines thereby reducing the wall-clock time for simulation. The number of threads spawned by the DES kernel is configurable. Each thread processes concurrent events (events with the same timestamp) in parallel without violating the causal constraints between events.

Table 8.1 The different agent instances used to develop the Eco-description used for case studies

Description of agent type	No. of instances	Total population	No. of countries
Bar-tailed godwit	4	40,000	18
Canada goose	16	231,700	5
Common crane	9	22,500	21
Eurasian widgeon[a]	3	1,296,000	17
Great knot	3	231,000	8
Mallard[a]	1	5,000	1
Razorbill	1	148,000	4
Red-breasted goose[a]	1	44,000	4
Red-crowned crane	1	15,000	4
Siberian crane	3	30,000	12
Yellow-billed duck	2	20,000	8
Total waterfowl flocks	44	4,371,000	40
Total poultry flocks	1,315	18,136,146,826	All
Total human groups	1,314	6,646,739,849	All
Total	2,673	24,787,572,675	All

Note that each agent is used to represent a group

[a] High risk waterfowl species (Hagemeijer and Mundkur 2006). The total of 44 waterfowl flocks with different migratory pathways were used. The total population column shows the sum of the populations of all agent instances in each category

8.3.1 Human-in-the-Loop Steering Module

In our current research, we have drawn inferences and conclusions from a variety of case studies conducted using a calibrated Eco-description. First an initial Eco-description was developed via SEARUMS using the modeling methodology described in Sect. 8.3. Table 8.1 lists the waterfowl species, including high risk species (Hagemeijer and Mundkur 2006), used to develop the Eco-description. The migratory flyways of the waterfowl and their population have been collated from data published by various organizations (CDC 2006; WHO 2006c; GROMS 2006; GLiPHA 2007, Hagemeijer and Mundkur 2006). For modeling and simulation purposes the dates for migration were approximated to the middle of the months reported in the statistics. Due to the significant variation in migration patterns the approximated migration dates are expected to have deviations of ±2 weeks which are accounted for through stochastic changes in migration dates each time a simulation is performed.

8.4 Methods

In our current research, we have drawn inferences and conclusions from a variety of case studies conducted using a calibrated Eco-description. First an initial Eco-description was developed via SEARUMS using the modeling methodology described in Sect. 8.3. Table 8.1 list the waterfowl species, including high risk species (Hagemeijer and Mundkur 2006), used to develop the Eco-description. The migratory flyways of the waterfowl and their population have been collated from data published by various organizations (CDC 2006; GLiPHA 2007; GROMS 2006; Hagemeijer and Mundkur 2006; USCB 2006; WHO 2006c). For modeling and simulation purposes the dates for migration were approximated to the middle of the months reported in the statistics. Due to the significant variation in migration patterns the approximated migration dates are expected to have deviations of ±2 weeks which are accounted for through stochastic changes in migration dates each time a simulation is performed. The radius of the waterfowl agents were computed using the population, density, and dispersion data obtained from published statistics (GROMS 2006; Hagemeijer and Mundkur 2006).

The dispersion of poultry population in different continents has been approximated to circular regions with even density (GLiPHA 2007; Law et al. 2005; Booth 1997). Such a modeling approach is commonly used in spatially explicit ecological models (Booth 1997; Hare and Deadman 2004; Law et al. 2005; Winston 1994; Wolfram MathWorld 2006). Global poultry and human population density data have been collated from statistics published by national organizations and government databases (GLiPHA 2007; SEDAC 2007; USCB 2006). As shown in Table 8.1, our model includes the complete human population (approximately 6.646 billion) humans represented by 1,314 agents. On an average, each human agent models 5.058 million humans living in a contiguous circular region. However, the precise population represented by an agent varies depending on the density of the region it models. Agents modeling dense metropolitan areas have higher human populations while agents modeling rural areas of the world have lower population. In contrast, the radius of all the human agents in the model is equal. The radius was computed using the grid size of gridded human population data from SEDAC (2007).

A similar strategy has also been applied to distribute the 18.136 billion poultry birds to 1,315 poultry agents as shown in Table 8.1. All the waterfowl agents have equal radius as determined from the grid size of the gridded poultry data obtained from GLiPHA (2007). However, the poultry population represented by each agent varies depending on the world region being modeled by the agent. Note that our Eco-description includes only a selected subset of the waterfowl as complete migration data is unavailable. However, to the best of our knowledge, it is the most comprehensive model of its kind reported to date. Furthermore, it can be readily extended to include additional waterfowl entities from other parts of the world.

8.4.1 Human-in-the-Loop Steering and Calibration

The first step in our study was to calibrate the model to ensure that it provides a sufficiently accurate representation of real-world epidemiology. The calibration was performed injunction with verification and validation of the model.

Table 8.2 Comparison of chronology of significant Real-world outbreaks as reported by WHO (2006) against the simulated outbreaks

Incident	Real-world date	Simulated date	Error (days)
Outbreak in Indonesia	23–Jan–2006	01–Jan–2006	22
Infection in Iraq/Iran	01–Mar–2006	25–Mar–2006	24
Infection in China	27–Apr–2006	02–Apr–2006	−25
Infection in Egypt	11–Oct–2006	14–Sep–2006	−27

The data was recorded after the model was calibrated

We verified the accuracy and fidelity of the aforementioned Eco-description by performing extensive HITL-simulations with initial source of infection set to outbreak in Indonesia, a notable epicenter of H5N1 epidemics (WHO 2006a). The Eco-description was interactively calibrated by suitably tuning the following attributes: start date for simulation, initial infection percentage, intra-flock disease spread rate, and inter-flock transmission mechanism. Note that we calibrated only the attributes that were indirectly derived from published statistics. We established validity of the Eco-description and SEARUMS by confirming that the timing and chronology of several outbreaks observed in the simulations correlate with the significant real-world incidents reported (WHO 2006a) as shown in Table 8.2.

It must be noted that the HITL-simulations steering played an important role in calibrating the model, particularly given its size and complexity. Using the validated and calibrated Eco-description we performed several case studies to analyze the spread and pandemic threat posed by avian influenza to US. Some of these case studies are discussed in Sect. 8.5.

8.5 Experiments

The calibrated Eco-description (see Sect. 8.4 for details) has been used to conduct several experiments to analyze the potential impact of avian influenza to poultry farming in the US. The study was conducted using a number of bio-simulations with three different experimental groups of migrating waterfowl. The three flocks were chosen based on their close proximity to known primary sites of disease outbreaks (Rao et al. 2009). The initial infection in each experimental group was varied for analysis.

Figure 8.6 illustrates one of the trans-Atlantic transmission pathways to the continental US. We observed that the spread was determined by migratory pathways and timelines of different species of waterfowl rather than initial infection percentages (Rao et al. 2009). One of the interesting observations is that our experiments correctly predicted an outbreak in the UK (WHO 2006c). The graph in Fig. 8.7 presents the impact of avian influenza outbreaks on poultry population in the continental US. Decrease in poultry population corresponds to H5N1 induced death and culling of birds to control the disease. Increase in poultry

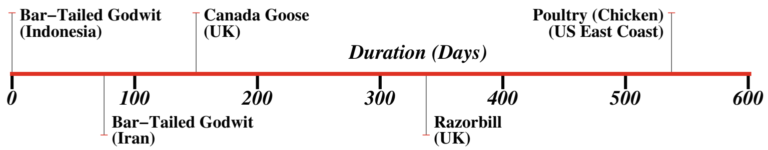
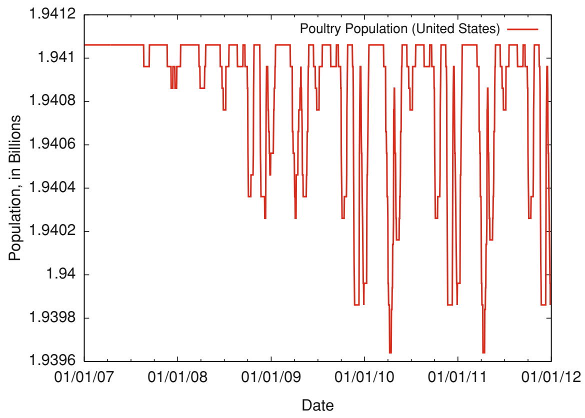


Fig. 8.6 Timelines for H5N1 infection spread from Indonesia (a known epicenter of H5N1 outbreaks) to US. The time chart also illustrates the epicenters of the intermediate infections

Fig. 8.7 Fluctuation in poultry population due to H5N1 outbreak which is controlled by mass culling of infected poultry



population reflects regeneration of poultry flocks after an outbreak. As illustrated by the graph, infections in poultry also follow a cyclic pattern that correlate with annual migration of waterfowl. The mortality figures can be translated to corresponding dollar figures for financial analysis. Additional case studies and experiments conducted using SEARUMS are discussed in the literature (Rao et al. 2009).

8.6 Conclusion

This article motivated the need for HITL-simulation steering (see Sect. 8.1.2) to effectively analyze the global epidemiology of emergent diseases. As discussed in Sect. 8.1, epidemiology is a vital discipline with far reaching impacts on human health and economics. Specifically, the article discussed the conceptual mathematical models (refer to Sects. 8.2.1 and 8.2.3) and their implementation in a custom Eco-modeling and HITL bio-simulation environment called SEARUMS. The article also discussed the software architecture of SEARUMS and its design in Sect. 8.3.

The procedure involved in utilizing SEARUMS for Eco-modeling and model calibration using HITL-simulation steering was discussed in Sect. 8.4. HITL-simulation steering played a vital role in enabling calibration of the model. The

HITL infrastructure was developed because conventional approaches for suitably seeding the model were unsuccessful despite hundreds of attempts and simulation runs. Furthermore, the ability to create checkpoints using the HITL infrastructure enabled rapid analysis of various scenarios.

The Eco-modeling and HITL-simulation infrastructure of SEARUMS has been used to model and analyze the global epidemiology of avian influenza. It must be noted that, even though the current emphasis of SEARUMS is on avian influenza, the underlying conceptual models and the software infrastructure are generic. Consequently, the concepts and software tools can be adapted for epidemiological analysis.

Some of the experiments conducted using SEARUMS were presented in Sect. 8.5 to illustrate the multi-disciplinary applicability of SEARUMS. Researchers, epidemiologists, and ornithologists can utilize HITL-simulations for rapid "what-if" types of analysis to study impacts of other factors influencing epidemics. It can be used to analyze other scenarios such as those simulated by Los Alamos National Laboratory (2006). SEARUMS and our Eco-description provide an excellent foundation for further enhancements. Note that use of SEARUMS does not require any special computing infrastructure or programming knowledge. Consequently, experts from multiple domains can collaboratively use SEARUMS to perform various types of analysis on a global scale, assess threats, and measure effectiveness of countermeasures. Our methodology and HITL-simulation environment will enable mankind to strategically invest precious time and resources to combat avian influenza, minimize its impacts on human life and global economy thereby averting a pandemic.

References

Anderson RM, May RM (1992) Infectious diseases of humans: dynamics and control. Oxford University, Oxford University Press

Bloch J (2001) Effective Java programming language guide, 1st edn. Pearson, Prentice Hall

Booth G (1997) Gecko: a continuous 2d world for ecological modeling. Artif Life 3(3):147–163

CDC (2006) Centers for disease control & prevention: avian Influenza: current situation. http://www.cdc.gov/flu/avian/outbreaks/current.htm

Daley DJ, Gani J (2001) Epidemic modelling: an introduction. Cambridge University Press, Oxford

Epstein JM (2009) Modelling to contain pandemics. Nature 460:687–687. doi:10.1038/nature460687a

Ferguson NM, Cummings DAT, Fraserl C, Cajka JC, Cooley PC, Burke DS (2006) Strategies for mitigating an influenza pandemic. Nature 442:448–452. doi:10.1038/nature04795

Flint SJ, Enquist LW, Racaniello VR, Skalka AM (2004) Principles of virology, 2nd edn. American Society for Microbiology (ASM), Washington, DC

Garten RJ, Davis CT, Russell CA, Shu B, Lindstrom S, Balish A, Sessions WM, Xu X, Skepner E, Deyde V, Okomo-Adhiambo M, Gubareva L, Barnes J, Smith CB, Emery SL, Hillman MJ, Rivailler P, Smagala J, de Graaf M, Burke DF, Fouchier RAM, Pappas C, Alpuche-Aranda CM, Lopez-Gatell H, Olivera H, Lopez I, Myers CA, Faix D, Blair PJ, Yu C, Keene KM, Dotson JP David, Boxrud D, Sambol AR, Abid SH, St George K, Bannerman T, Moore AL, Stringer DJ, Blevins P, Demmler-Harrison GJ, Ginsberg M, Kriner P, Waterman S, Smole S, Guevara HF, Belongia EA, Clark PA, Beatrice ST, Donis R, Katz J, Finelli L, Bridges CB, Shaw M, Jernigan DB, Uyeki TM, Smith DJ, Klimov AI, Cox NJ (2009) Antigenic and genetic characteristics of swine-origin 2009 A(H1N1) influenza viruses circulating in humans. Science 325(5937):197–201. doi:10.1126/science.1176225, http://www.sciencemag.org/cgi/reprint/325/5937/197.pdf

Gilbert N, Bankes S (2002) Platforms and methods for agent-based modeling. In: Proceedings of the National Academy of Sciences of the USA, vol 99, pp 7197–7198. doi:10.1073/pnas.072079499

GLiPHA (2007) Global livestock production and health atlas (GLiPHA): animal production and health division of food and agriculture organization of the UN. http://www.fao.org/ag/aga/glipha/index.jsp

GROMS (2006) Global register of migratory species: summarising knowledge about migratory species for conservation. http://www.groms.de/

Gu W, Vetter J, Schwan K (1994) An annotated bibliography of interactive program steering. ACm SIGPLAN notices 29(9):140–148. doi:10.1145/185009.185038

Hagemeijer W, Mundkur T (2006) Migratory flyways in Europe, Africa, and Asia and the spread of HPAI H5N1. In: International scientific conference on avian influenza and wild birds, FAO and OIE, Rome, Italy. http://www.fao.org/ag/againfo/subjects/en/health/diseases-cards/conference/documents/hagemeijer-mundkur.pdf

Halloran ME, Ferguson NM, Eubank S, Ira M Longini J, Cummings DAT, Lewis B, Xu S, Fraser C, Vullikanti A, Germann TC, Wagener D, Beckman R, Kadau K, Barrett C, Macken CA, Burke DS, Cooley P (2008) Modeling targeted layered containment of an influenza pandemic in the US. Proc Natl Acad Sci USA 105(12):4639–4644. doi:10.1073/pnas.0706849105, http://www.pnas.org/cgi/doi/10.1073/pnas.0706849105

Hare M, Deadman PJ (2004) Further towards a taxonomy of agent-based simulation models in environmental management. Mathematics and computers in simulation 64(1):25–40. doi:10.1016/S0378-4754(03)00118-6

Hufnagel L, Brockmann D, Geisel T (2004) Forecast and control of epidemics in a globalized world. Proc Natl Acad Sci USA 101(42):15124–15129. doi:10.1073/pnas.0308344101, http://www.pnas.org/content/101/42/15124.abstract

Kilpatrick AM, Chmura AA, Gibbons DW, Fleischer RC, Marra PP, Daszak P (2006) Predicting the global spread of H5N1 avian influenza. Proc Natl Acad Sci USA103(51):19368–19373. doi:10.1073/pnas.0609227103

Lamport L (1978) Time, clocks, and the ordering of events in a distributed system. Commut ACM 21(7):558–565

LANL (2006) Los Alamos National Laboratory: avian flu modeled on supercomputer, explores vaccine and isolation options for thwarting a pandemic. http://www.lanl.gov/news/index.php?fuseaction=home.story&story_id=8171

Law R, Dieckmann U, Metz JA (2005) The geometry of ecological interactions: simplifying spatial complexity. Cambridge University Press, New York

Longini IM, Nizam A, Xu S, Ungchusak K, Hanshaoworakul W, Cummunings DAT, Halloran ME (2005) Containing pandemic influenza at the source. Science 309(5737):1083–1087. doi:10.1126/science.1115717

Normile D (2006) Avian influenza: evidence points to migratory birds in H5N1 spread. Science 311(5765):1225. doi:10.1126/science.311.5765.1225

Railsback SF, Lytinen SL, Jackson SK (2006) Agent-based simulation platforms: review and development recommendations. Simulation 82(9):609–623. doi:10.1177/0037549706073695

Rao DM, Chernyakhovsky A (2008) Parallel simulation of the global epidemiology of avian influenza. In: Proceedings of the 2008 winter simulation conference, pp 1583–1591

Rao DM, Chernyakhovsk A, Rao V (2007a) SEARUMS: studying epidemiology of avian influenza rapidly using simulation. In: Proceedings of ICMHA'07. Berkeley, pp 667–673

Rao DM, Martin Stieger DDH Jr, Johnson CB, Kidambi P, Narayanan S (2007b) Design and implementation of virtual and constructive simulations using open EAAGLES. Linus Publications, New York

Rao DM, Chernyakhovsky A, Rao V (2009) Modeling and analysis of global epidemiology of avian influenza. Environ Model Softw 24(1):124–134. doi:10.1016/j.envsoft.2008.06.011

Russell CA, Jones TC, Barr IG, Cox NJ, Garten RJ, Gregory V, Gust ID, Hampson AW, Hay AJ, Hurt AC, de Jong JC, Kelso A, Klimov AI, Kageyama T, Komadina N, Lapedes AS, Lin YP, Mosterin A, Obuchi M, Odagiri T, Osterhaus ADME, Rimmelzwaan GF, Shaw MW, Skepner E, Stohr K, Tashiro M, Fouchier RAM, Smith DJ (2008) The global circulation of seasonal influenza A (H3N2) viruses. Science 320(5874):340–346. doi:10.1126/science.1154137

SEDAC (2007) SocioEconomic Data and Applications Center (SEDAC): gridded population of the world. http://sedac.ciesin.columbia.edu/

Smith GJD, Vijaykrishna D, Bahl J, Lycett SJ, Worobey M, Pybus OG, MaSK Cheung CL, Raghwani J, Bhatt S, Peiris JSM, Guan Y, Rambaut A (2009) Origins and evolutionary genomics of the 2009 swine-origin h1n1 influenza a epidemic. Nature 459:1122–1125. doi: 10.1038/nature08182

Solow AR, Smith WK (2006) Using markov chain successional models backwards. J Appl Ecol 43(1):185188. doi:10.1111/j.1365-2664.2005.01127.x

Tobias R, Hofmann C (2004) Evaluation of free java-libraries for social-scientific agent based simulation. J Artif Soc Soc Simul(JASS) 7(1). http://jasss.soc.surrey.ac.uk/7/1/6.html

USCB (2006) U.S. Census Bureau: 100 largest counties. http://www.census.gov/popest/counties/CO-EST2006-07.html

WHO (2005) World Health Organization: ten things you need to know about pandemic influenza. http://www.who.int/csr/disease/influenza/pandemic10things/

WHO (2006a) World Health Organization: avian influenza: updates. http://www.who.int/csr/disease/avian_influenza/updates/en/

WHO (2006b) World Health Organization: H5N1 avian influenza: timeline. http://www.who.int/entity/csr/disease/avian_influenza/timeline.pdf

WHO (2006c) World Health Organization (WHO): avian influenza. http://www.who.int/csr/disease/avian_influenza/en/

WHO (2007a) A description of the process of seasonal and H5N1 influenza vaccine virus selection and development. http://www.who.int/entity/csr/disease/avian_influenza/Fluvac-cvirusselection.pdf

WHO (2007b) World Health Organization: avian influenza: situation updates. http://www.who.int/entity/csr/disease/influenza/H5N1-9reduit.pdf

Winston WL (1994) Operations research (applications and algorithms), 3rd edn. Duxbury Press, Belmont

Wolfram MathWorld (2006) Mercator projection. http://mathworld.wolfram.com/MercatorProjection.html

Chapter 9
Aiding Understanding of a Contested Information Environment's Effect on Operations

Michael W. Haas, Robert F. Mills and Michael R. Grimaila

The essence of winning and losing is in learning how to shape or influence events so that we not only magnify our spirit and strength but also influence potential adversaries as well as the uncommitted so that they are drawn toward our philosophy and are empathetic towards our success.

Col. John Boyd (Dewar et al. 1996)

Abstract The operations of government, industry, the military, academia, and even personal activity can be negatively affected by information attacks on and through cyberspace. Modeling and simulation can be used to increase the understanding of potential effects these attacks may generate and guide the development of contingency planning. The increased understanding and more comprehensive and focused contingency planning will enhance the ability of organizations to assure their operation or mission, when operating in a contested information environment. This enhanced mission assurance will increase the overall national security and the deterrence against the use of information attacks in the future. This chapter pulls these concepts together and develops requirements for modeling and simulation capabilities to enhance mission assurance.

M. W. Haas (✉)
Human Effectiveness Directorate, 711th Human Performance Wing,
Air Force Research Laboratory, Wright-Patterson AFB, USA
e-mail: michael.haas@wpafb.af.mil

R. F. Mills
Department of Electrical and Computer Engineering,
Air Force Institute of Technology, Wright-Patterson AFB, USA

M. R. Grimaila
Department of Systems and Engineering Management,
Air Force Institute of Technology, Wright-Patterson AFB, USA

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_9,

9.1 Introduction

Operations of modern civilian and military organizations depend more highly now, than ever before, on the ability to process a large quantity of data and to quickly, accurately, and securely exchange data internally and externally with their organization. Increases in data processing and data exchange capability have been enabled by the development of a world-wide networking of computers and automated systems, the global Internet. The foundational concepts and initial prototypes were a product of the US Department of Defense but overwhelmingly have been used for peaceful purposes (Libicki 2007). Cyberspace is defined by the US Department of Defense as a global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the Internet, telecommunications networks, computer systems, and embedded processors and controllers. (Castelli 2008). Cyberspace provides an information-rich domain to individuals and organizations around the globe enabling higherlevels of communication, knowledge exploration, and collaboration not achievable even a decade ago. These advances have brought with them both offensive capabilities and defensive vulnerabilities. Thus, cyberspace has become a contested environment.

Cyberspace is also an *asymmetric* contested environment. The asymmetry results from a characteristic of cyberspace; that is, it is less costly to develop and employ offensive capability than provide a defensive capability against those same offensive capabilities. This asymmetry has enabled hostile activity to grow and be focused toward organizations and nation states that depend most heavily on information. Hostile activity occurs in, and through, cyberspace almost continually and the sophistication of these hostile activities continues to rise with time (Verizon Business Risk Team 2009). These trends in hostile activities have made cyberspace, including the information capability in which organizations rely, an increasingly contested environment.

In spite of the contested nature of the information environment, decision-makers and problem-solvers operating within this information environment must maintain their organization's operation under the threat of, and during the execution of, information attacks targeted toward them. In other words, the organization must be able to understand the possible impacts to the organization's operations, adaptively defend against the threat, and compensate for degradations in the information environment. This chapter will explore modeling and simulation constructs associated with developing and maintaining the understanding needed to accomplish these difficult, but necessary, undertakings. Additionally in this chapter, the integration of modeling and simulation capability into military planning and operational cycles will be discussed as a template for integration into large organizations in general.

9.2 Information Operations

The information environment is utilized by decision-makers and problem-solvers at all levels of responsibility and in all segments of society; government, military, industrial, academic, and personal. Intentional shaping of the information environment is also performed by organizations and individuals in the society with variance in the number of individuals affected by the shaping. Traditional examples of organizations and individuals capable of shaping large segments of the information environment, and large numbers of individuals, are the news media, the broadcast media, and, national leaders such as the President or a State Governor. More recently, cyberspace has provided additional capability to shape the information environment. Web sites that become popular, YouTube videos with high number of viewers, or blogs with large number of participants are all examples of these more recently available modes of information environment shaping.

Cyberspace has provided a greatly increased number of sources that have an ability to shape the information environment. Because of this increase, it is difficult to imagine the content of the entire information environment as a whole. It is more constructive to think about the information content as a collection of topics, for which multiple sources may be capable of contributing to any given topic or topics. In addition, for any given topic, there are a number of consumers that may be associated with that topic. The number of sources and consumers is dynamic across time as are the interactivity level of both sources and consumers. The content of the information environment for a given topic at a given time could be considered as an information set resulting from the contribution of its many sources up to that point in time. This characteristic of the information environment is not significantly different from thinking of the entirety of the air, space, or maritime domains. At any given moment in time, an organization may only be interested in the security of a portion of that domain, and would not ever attempt to maintain the security of the entire domain. However, the information domain does differ from these other domains in that the security segmentation of the non-cyber domains is a geo-spatial construct. A geo-spatial construct cannot be used to segment cyberspace for security purposes.

There are topics however, for which there is a purposefully limited number of sources. These limited sources are tightly controlled as are the consumers of that information. Examples of these would be topics that are classified for national security reasons, proprietary industrial information, trade secrets, or personal financial information.

Both open topics and limited topics are utilized by decision-makers and problems solvers and can be the focus of information operations. It is the decision-makers and problem-solvers that are the targets of these information operations and not the information technology infrastructure, functionality or contents. This can best be seen in the US Department of Defense Joint Publication (JP) 3–13 and is discussed below (Joint Publication 2006).

A model utilized in Joint Publication 3–13, as well as in many other locations when discussing decision-making, is John Boyd's Observe, Orient, Decide, and

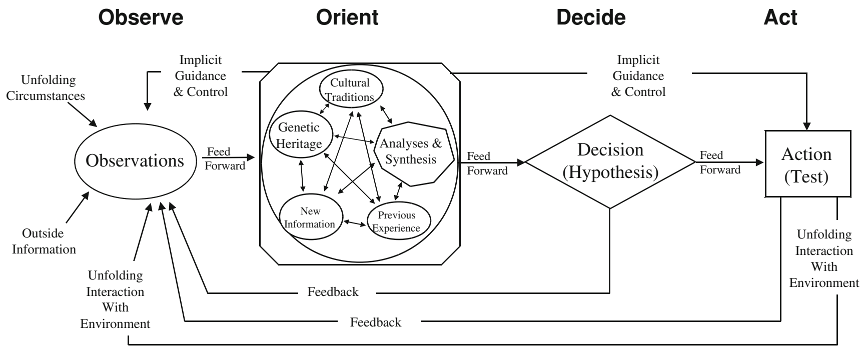


Fig. 9.1 Boyd's observe, orient, decide, act (OODA) loop (Boyd 1987)

Act (OODA) loop. The OODA loop is shown in its expanded form in Fig. 9.1. The OODA loop depicts a closed loop between the environment in which the decision-maker is operating within the perceptual and cognitive activities of the decision-maker, and finally the action taken by the decision-maker that then affects the environment. Also depicted, in the feed forward part of the-loop on the top, is a representation of "implicit guidance and control", that Boyd theorized represented instinctual reactions that could occur more quickly than calculated decision-making and could be learnt over time.

The OODA loop was first described by John Boyd, a fighter pilot, as a way to model the decision-making of both friendly and adversary fighter pilots. Boyd's most simple application of the OODA Loop model in combat was of two OODA loops operating simultaneously, in the same airspace, one representing a friendly fighter pilot and the other representing an adversary fighter pilot. In Boyd's terminology, winning the advantage over the adversary was *getting inside the adversary's OODA loop*. This terminology is still used today. It is the foundation for Waltz's basic model of the information processes in a conflict between a single attacker and a single defender (Waltz 1998). The OODA terminology is also used in Joint Publication 3–13 in terms of an over-arching goal of employing information operations techniques.

The OODA loop is utilized today to represent not only the decision-making of an individual, but also the behavior of a group of individuals working on a single decision or combinations of tasks in which multiple decisions are embedded within each task. Ullman (2007) postulated that decision-makers get stuck at observe and orient when sufficient uncertainty exists. This property of decision-making could be used to guide an information operations technique; that technique being used to slow down, or even stop, the decision process of an adversary keeping them from acting on previously gathered information.

The US military describes Information Operations in Joint Publication 3–13 as "the integrated employment of electronic warfare (EW), computer network operations (CNO), psychological operations (PSYOP), military deception (MILDEC), and operations security (OPSEC), in concert with specified supporting and related capabilities, to influence, disrupt, corrupt or usurp adversarial human and automated decision-making while protecting our own". The targeting of decision-making is an important point and one that is not intuitive. While the target of an information operation is a decision-maker, the desired effect must be accomplished through the manipulation of information sources, the information content itself, or indirectly through affecting the information interpretation by the decision-maker. Effects to be achieved by information operations must use information techniques as an indirect path to the cognitive processing of the decision-maker or problem-solver. The application of information techniques is difficult as the coordination of techniques is critical to the effectiveness of the information operation over the duration required and over the decision-makers targeted. In addition, the complexity of planning and executing an information operation is increased by the necessity of controlling and monitoring as many potential sources of information as possible that may impact the targeted decision-maker or problem-solver. This control and monitoring must be maintained over the duration of the operation. Adaptive information techniques must also be employed when the desired effect is not being achieved due to factors outside the control of the operation itself. Maintaining the information environment to any given state, and maintaining the desired decision-making effect, increases in difficulty with increases in the operation duration. Complexity also increases with the planned or unplanned generation of cascading effects. Cascading effects are derivatives of the employed information techniques generating the initially planned, or subsequently generated unplanned effects on decision-makers and problem-solvers.

Joint Publication 3–13 represents the information environment as three dimensions in which information operations can have effect, the physical-dimension, the information dimension, and the cognitive dimension (Joint Publication 2006). Alberts et al. (2001) also use three spaces to divide information age warfare. Within the physical, information, and cognitive dimensions, the intended effect of the information operations may include the detection, deterrence, deception, disruption, defense, denial, or defeat of an adversary. Information operations may be conducted during peacetime as well as during armed conflict. Additionally, information operations may be conducted in times of stress, when armed conflict may not have been initiated or may have paused, in the hope of eliminating escalation. When two or more organizations are attempting to operate freely in the information environment by reducing one another's ability to operate freely, those organizations can be involved in information warfare.

9.3 Security of the Information Environment

In the past, information operations during times of war have been called information warfare. As the definition of war has evolved through the years, moving from a strictly legal definition to a word indicating a form of "intense" conflict, so with the definition of information warfare. Unfortunately, this has led to many definitions of information warfare existing in the literature. For example, Martin Libicki develops information warfare into seven categories; command and control-based warfare, intelligence-based warfare, electronic warfare, psychological warfare, hacker warfare, economic information warfare, and cyber warfare (Libicki 1996). The US military has, in the past, defined information warfare in various ways. A typical definition would include the actions taken to preserve the integrity of one's own information systems from exploitation, corruption, or disruption, while at the same time exploiting, corrupting, and destroying an adversary's information systems, and in the process achieving an information advantage in the application of force. In today's military vernacular, the term *information operations* is used to cover all activities in and through the information environment independent of the state of conflict that exists.

Information warfare, and control over the contested state of the information environment, has always been a recognized component of military operations. As an example, Sun Tzu, perhaps in the fifth century BC, wrote "All warfare is based on deception. Hence, when able to attack, we must seem unable; when using our forces, we must seem inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near." Tzu (1994) as translated from the Chinese in 1910 by Lionel Giles. However, information warfare has taken on a much greater significance beginning in the 1990s according to Wood (1995). This is probably due to the recognition of the potential usefulness of the non-lethal nature of information operations if they can eliminate destruction, and ultimately re-building effort. It could also be due, in part, to the fact that many nation states and organizations are increasingly dependent on information capability, they are also the most vulnerable to information warfare attacks, and information warfare can be an asymmetric form of warfare with little cost to the offensive side. This argument is reflected in many sources, including a recent book by Paul (2008).

The distinguishing characteristics of information warfare include its non-kinetic nature and it is focused on targeting the decision-making and problem-solving capability of the adversary.

9.4 Information Operations in and Through Cyberspace

Cyberspace is a domain in which technologies can be utilized to employ information operations and is heavily utilized by a large segment of the world's population. Cyberspace is a contested domain, a part of the overall information environment. Cyberspace is an increasingly important domain for the employment of offensive information operations as well as an increasingly important domain to defend.

The US Department of Defense's current definition of cyberspace, from May 2008, is:

> Cyberspace is a global domain within the information environment consisting of the interdependent network of information technology infrastructures, including the Internet,

telecommunications networks, computer systems, and embedded processors and controllers.

In a September 2008 memorandum, the US Department of Defense defined cyberspace operations in this way.

Cyberspace operations: The employment of cyber capabilities where the primary purpose is to achieve military objectives or effects in or through cyberspace. Such operations include computer network operations and activities to operate and defend the Global Information Grid.

This same memorandum also stated:

The use of cyber capabilities to create desired effect is essential to the nature of a cyber operation: Cyberspace operations require the use of cyber capabilities (e.g. computers, software tools, networks, cyber forces). Operations that may cause effects in cyberspace (e.g. electronic warfare, psychological operations) that do not employ cyber capabilities should not be considered cyberspace operations. Cyberspace operations can cause effects in any domain.

9.5 Heightened Focus on Cyberspace Operations

The opportunity to conduct information operations in and through cyberspace increases as the number of sources, the number of consumers, and the volume of data and information within cyberspace increases. The viral increases in the users, and uses, of social engineering web services such as MySpace and Facebook is a clear indication of these ever-increasing opportunities. Technologically, computer network operations may be the fastest growing component of information operations and in the US, has prompted the creation of a new sub-unified command under USSTRATCOM called USCYBERCOM. USCYBERCOM is commanded by General Alexander, who is also the Director of the National Security Agency. The creation of USCYBERCOM is an indication of how important the US views its cyberspace capability, how seriously the US views the threat from cyberspace, and enables an efficient coordination and control of operations across the service component's networked equipment and cyberspace defensive, offensive, and exploitation capability. The cyberspace capability, typically described as cyber network defense (CND), cyber network attack (CNA), and cyber network exploitation (CNE), is integrated with the kinetic offensive and defensive military capability to provide integrated national security across all of the war fighting domains; land, air, space, maritime(surface and sub-surface), and cyberspace.

A very significant change has occurred in the scope of the cyber warriors' role. Changes in the US Department of Defense joint doctrine, JP 3–13, as well as service specific doctrines, such as air force doctrine document (AFDD) 3–12, now indicate that cyber warriors are not only responsible for maintaining the operation and capability of the network itself, but are also responsible for assuring that the military mission operating on the network continues to execute in the information contested environment brought about by adversarial attacks affecting the information system (Joint Publication 2006; Air Force Doctrine Document 2010). This new responsibility is to be borne not only by the cyber warriors, but also by the war fighters conducting missions in the other domains. Closer coordination between the cyberspace and other domain war fighters will be required to achieve the maximum level of mission assurance. Increased understanding of how operations in cyberspace, both friendly and adversary, will affect the friendly mission will be required. Requirements for additional modeling and simulation will increase for both mission planning and mission execution phases of operations. More timely and accurate interpretation, as well as broad dissemination, of expanded modeling and simulation results will also need to be developed.

9.6 Mission Assurance, Deterrence, and Increased National Security

By developing and integrating modeling and simulation capability that reflects the effects of cyber attacks (information operations employed through cyberspace) into the military operational cycles of planning and execution, enhanced mission assurance can be achieved. Enhanced mission assurance can deter an adversarial decision-maker from employing offensive information warfare operations by greatly reducing the operation's potential affect on the mission performance.

9.7 Mission Assurance: Linking Computer Network Operations with Missions

For a joint operations perspective, military organizations are segmented into different groups based upon their primary function. Previous research at the Air Force Institute of Technology and the Air Force Research Laboratory has identified a chasm that exists between the J-3, which is responsible for the planning and conduct of military operations (to include oversight of policies, intelligence, manpower, communications and logistics operations) and the J-6, which is responsible for all aspects of the Command, Control, Communications, and Computer (C4) systems (Fry 2009; Mullen 2009). Each element within the J-3 community views the J-6 community as a utility which provides the C4 capability to operations. Conversely, the J-6 community mission is to provide C4 capabilities to all its customers. When an information incident occurs in the network, the J-6 community is charged with aggregating and reporting the operational impacts from each affected unit, but especially from those who are conducting operations in the J-3 community. As a result, many of the existing efforts to provide situational awareness focus on providing network personnel with an understanding from the operational community about the criticality of network and systems. In contrast, it may be that to attain mission assurance, focus should also be placed on developing techniques and tools to provide the J-3 operational community with a better understanding of how their mission is impacted by an information incident experienced within the J-6 environment.

Tinnel et al. (2003) recognized that a dichotomy exists in the perception of the value of the information infrastructure between network defenders and operations personnel in military operations. Network defenders are focused on assuring the health of the networked information infrastructure with a limited view of the operational importance of the missions supported. In contrast, operations personnel tend to be focused upon their own missions with little understanding of how the missions depend upon the cyberspace infrastructure. However, both communities are inherently linked by their symbiosis. Stated in another way, network operations personnel focus upon maintaining the health and safety of the network and information systems while the mission operations personnel, who inherently rely upon the network and information systems, focus on assuring their mission operations through command decision-making. Tinnel et al. (2003) recognized that this gap as a key limitation to network defenders. However, a greater negative impact can be experienced by the operational community because they are unaware of the significance of an information incident in terms of the ability to complete tasks in support of their stated mission (Grimaila et al. 2009a). Properly managing operational risk in cyberspace requires the ability to maintain a real-time awareness of the "state" of the information resources used to meet mission requirements.

Many barriers are identifiable that hinder organizations in attaining this goal:

- Failing to collect and maintain critical information asset inventory (Fortson and Grimaila 2007)
- Failing to maintain critical information asset inventory (Fortson and Grimaila 2007)
- Focusing exclusively upon systems instead of information (Sorrels et al. 2008)
- Failing to appreciate the value of an information asset (Hellesen et al. 2008)
- Broadcast notification following an information incident (Grimaila et al. 2009a)
- Failure to notify all downstream-dependent entities in a timely manner (Grimaila 2008)
- Information filtering in the notification chain (Grimaila et al. 2009b)
- Lack of knowledge continuity (Grimaila et al. 2009c)
- Lack of relevant notification (Grimaila et al. 2009c)

It is possible that modeling and simulation can be used to provide operational units with the ability to understand their resource dependencies and the criticality of their own mission in terms of the effects of current and future information attacks in and through cyberspace. In essence, the mapping of information resources to operations, and operational entities, is manifested in the model of the operation itself and the simulation of that model enables the linkages to become dynamic. It would be the responsibility of the modelers to correctly map the information resources to the operational entities and thus, reduce the burden to both the operational organizations, and the command and control organizations, to create and maintain a static database of resource association mappings. In achieving this goal, individual operational units will obtain a more relevant, mission centric understanding of their own mission capability, the ability to take contingency actions where appropriate, and the ability to feed mission capability status information to other organizations within the theatre to help raise mission situational awareness and enable the development of a global strategic common operating picture (COP).

9.8 Deterrence of Information Attacks in and Through Cyberspace

It is encouraging to see an increased focus in joint and air force doctrine being devoted to cyber infrastructure resiliency and recovery efforts, in addition to the usual heavy emphasis on network protection. However, there are several foundational research themes which must be addressed in depth to support the implementation of a cyberspace deterrence capability. Deterrence of attacks in and through cyberspace presents some significant challenges, as noted by Vice Admiral Carl V. Mauney, Deputy Commander, USA Strategic Command:

> We face emerging forms of 21st Century warfare—transnational terrorism, cyber warfare, and counter-space warfare—which we have little experience in deterring. We need to think carefully about how deterrence will or will not apply to these threats and we need to tailor our deterrent strategy and associated capabilities accordingly. I believe deterrence does have a critical role to play in these threats (Mauney 2009).

Libicki (2009) raised several hard questions about deterrence in cyberspace: How can we differentiate between spying and an attack? Is spying cause for retaliation? Can we determine who conducted the attack? Can we actually retaliate over a given offense or impose costs for cyberspace attacks? How do we avoid escalation? Does deterrence necessarily imply attacking and response in kind? How do we deter against threats that are vague and nonspecific? (Libicki 2009).

Effects caused by offensive cyber actions can be classified as being made in the physical domain and in the cognitive domain. Examples of physical domain effects may range from shutting down electrical power grids to physical damage by causing a device to operate outside its normal operating parameters. From a modeling and simulation perspective, these effects can be evaluated on test hardware and software that replicate, as closely as possible, the hardware and software environment being targeted. It is much more difficult to evaluate the cognitive domain effects. Thus, the predictive power of cognitive domain effects, and resultant cascading cognitive domain effects, is currently very limited. Cognitive domain effects are a function of many variables, such as culture, perceived conditions of one's environment, the perceived target of the attack, level of belief in the knowledge of the attacker's identity, content of any present internal media coverage, and reaction by governmental, secular, or religious leaders.

In general, deterrence is achieved by convincing an adversary not to act in a manner that is undesirable to the US and its coalition partners. More specifically, the US and its coalition partners must "decisively influence the adversary's decision-making calculus in order to prevent hostile actions against US vital interests (United States Strategic Command 2006)". This implies that there must be an understanding of the cause and effect relationship between a given action to be undertaken by the USA and/or its partners, and the subsequent "decision-making calculus" undertaken by an adversary. Because of the asymmetric nature of cyberspace, there are many potential adversaries, including nation states, non-state actors, organized criminal activities, and other interconnected ad hoc groups. As such, there are myriad interrelated responses which could be triggered by any action, or set of sequential actions. In addition, it is not beyond the realm of possibility that a set of actions could induce one or more of these inter-related nation states, non-state organizations, or ad hoc groups to move from a neutral mindset to an adversarial mindset adding complexity, and potentially deepening a conflict. This is true of either defensive or offensive actions, if made public. Development of a robust understanding of the cause and effect linkage between defensive and offensive cyber actions undertaken for deterrence and resultant influence on adversaries, and potential adversaries must be accelerated and maintained at a high level to keep pace with the advanced technologies in the cyber domain. This development will reduce the potential of deepening future conflicts with mistimed cyber actions and result in a higherlevel of security.

US military operations and activities contribute to deterrence by affecting the adversary's decision calculus elements in three ways: *imposing costs, encouraging restraint, and denying benefits* (United States Strategic Command 2006). Deterrence is successful when the perceived costs incurred by an adversary outweigh the perceived benefits in regard to the consequences of restraint. Deterrence fails when an adversary perceives that the benefit of taking an action outweighs any associated costs and then commits that action.

Traditional nuclear deterrence strategy tended to focus on imposing costs—i.e., make the adversary expend a lot of resources in order to achieve the desired goal. If an adversary perceives that their preparations for attack are likely to be detected and preempted by the US, they may be deterred from initiating the attack. The benefit of conducting the attack is therefore denied by preemption. This "detect and preempt" strategy is less effective for cyberspace due to the very nature of the domain and the compression of time and space (Beeker et al. 2010). Key identifiers to predict, detect, track, and describe an incoming cyber attack are minimal compared to the physical world. There are no missile plumes to detect in cyberspace, and traditional boundaries are very blurred or nonexistent.

Transactions in and through cyberspace are built on computer communication protocols and trust. The anonymous nature of the Internet and the sheer volume of network traffic make discrimination between legitimate traffic and an attack difficult. When a packet shows up at the firewall, it is extremely difficult to determine whether it is malicious…and if it is malicious, the attack is already underway. This greatly complicates the deterrence problem: Who is attacking, for what purpose, and how sure are we? How can we respond effectively if we do not know who did what? (Vijayan 2005).

"Detect and preempt" is externally focused on the adversary and seeks to impose costs for doing what we don't want. This approach will not work for all potential adversaries—there are too many of them, and a one-size-fits-all strategy will not work for deterrence in cyberspace. An alternative strategy is to look inward and mitigate our own risks and dependencies on cyberspace. By doing this, we deny adversaries from deriving any benefit from their actions. For example, consider how the US military deters adversaries from using chemical or biological weapons. We equip our people with special equipment and train them to carry out their missions despite the use of chemical weapons on the battlefield. By doing this we achieve two objectives. First, we communicate to the adversary that we are prepared to operate in the event those weapons are used. Second, should the adversary use those weapons, our training ensures that we can operate safely and accomplish the mission regardless. This is really what mission assurance and "fight through" are all about (Beeker et al. 2010).

Air force doctrine document 3–12 says "Mission assurance ensures the availability of a secured network to support military operations by assuring and defending the portion of the network directly supporting the operation" and that "These actions may include focusing the attention of network defense assets on the slice of the network supporting the operations and conducting operations to ensure that no threats are resident on the network (Air Force Doctrine Document 2010)". Network maintainers—the traditional J-6 community—at all levels are primarily concerned with keeping the network operational. This activity is necessary but not sufficient, and an often overlooked part of the mission assurance puzzle is business continuity planning (BCP) or continuity of operations planning (COOP). BCP—the equivalent of training with chemical/biological weapons equipment—mitigates operational risks when confronted with disaster, cyber attacks, or other serious events. Mission assurance planning is not new and should be included in any organization's continuity plans, especially those dealing with critical processes and functions. This planning requires significant introspection by the user community and should not be left to the "IT guys" to sort out. Only the end users can best understand and appreciate the context of the information traversing the network.

Unfortunately, BCP planning is often lacking in cyber infrastructure) (Vijayan 2005; U.S. Government Accountability Office 2006). The 2009 national infrastructure protection plan and its underlying sector-specific plans attempted to remedy this situation (Department of Homeland Security 2009), but the primary emphasis in these documents continues to be on protection of cyber assets rather than recovery and remediation. As a result, these documents contain little detail and guidance for generating robust continuity plans.

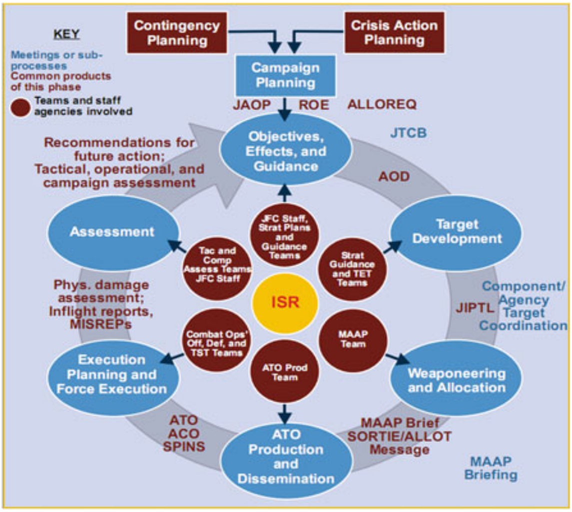


Fig. 9.2 Air tasking cycle (Joint Publication 2010)

9.9 Modeling and Simulation Requirements for Increased Mission Impact Understanding

Recently, Paulhamus et al. (2009) published an article in which they analyzed the effect of information attack on air and missile defense performance. This effort, in some ways, parallels the modeling and simulation needed to understand the effects that attacks through cyberspace may affect the planning or executing of a military mission (Paulhamus et al. 2009). In this article, a five step process was utilized to analyze the effect of information operations on a military capability, in this case air and missile defense. The Paulhamus et al. (2009) five steps are described below:

1. Select an operational scenario
2. Identify potential information attacks
3. Simulate the information attacks in a force-level context
4. Analyze modeling and simulation results to quantify impact of information attacks
5. Leverage analysis results to develop a metric framework that links low-level 10 metrics to high-level airborne missile defense performance metrics

An analogy can be drawn between what may be needed in an operational setting, using modeling and simulation to improve the level of mission assurance, and the Paulhamus five step methodology. In Fig. 9.2, the results of the modeling and simulation effort could be included in the MAAP briefing as well as input to the execution planning and for execution bubble on the lower left side of Fig. 9.2.

Step one of the Paulhamus methodology is the selection of an operational scenario which can be considered in an operational setting to be the "mission". Step two is the identification of the attack vectors, which, in the operational setting, is typically information obtained and disseminated by the intelligence function based on intelligence, surveillance, and reconnaissance activities coupled with analysis of those data by the intelligence community. The remaining three steps in the Paulhamus methodology are the modeling and simulation activity. The results of the modeling and simulation activity in the Paulhamus methodology are quantitative measures of mission performance, specifically the number of adversary missiles which are not adequately defended against and which may impact on their targets.

Operationally, the results of the modeling and simulation activities must be inserted into the mission planning and execution cycle to enable their utilization by the war fighter, either in developing alternative plans and/or to develop real-time, or near real-time, mission execution work-arounds to cope with the potential for operations with degraded systems and capability. The insertion of the modeling and simulation results may be best described visually by depicting them in a hypothetical mission planning and execution cycle. The need to consider effects generated in and through cyberspace is emphasized in Joint Publication 3–30 (Joint Publication 2010).

> Commanders at all levels must consider how our space and cyberspace capabilities enhance the effectiveness and execution of joint air operations. It is important to understand that in today's complex operational environment, adversary actions can be conducted on, from, within, and outside of the operational area, all with potentially global impacts and influence (Joint Publication 2010).

In Joint Publication 3–30, there are several references to the need to consider, and the need to coordinate information operations and cyberspace operations. However, there is no mention in the planning process of developing contingency plans to fight-through any potential adversarial-generated cyberspace attacks. This is not surprising as till date, there has been no technological solution available to capture the system-of-system effects that attacks in and through cyberspace could produce in a manner that could be inserted into the operational cycles of planning and execution. This is partially due to the complexity of forecasting how potential cyber attacks could propagate through a military operation and all of its associated automation, platforms, and decision-makers as the propagation process has elements that would occur very quickly, elements that would occur slowly, and the combinatorial interactions with friendly defensive actions would be difficult to comprehensively evaluate. The number of research questions to be answered to develop this capability is large but the vision of the end-product, its utilization, and its potential for significantly increasing fight-through capability cannot be in doubt.

9.10 Elemental Components of a Modeling and Simulation Capability

The most elemental components of a model representing the information environment and modeling the effects of information warfare attack are the content from the datum source, the model of the decision-maker, and the observable

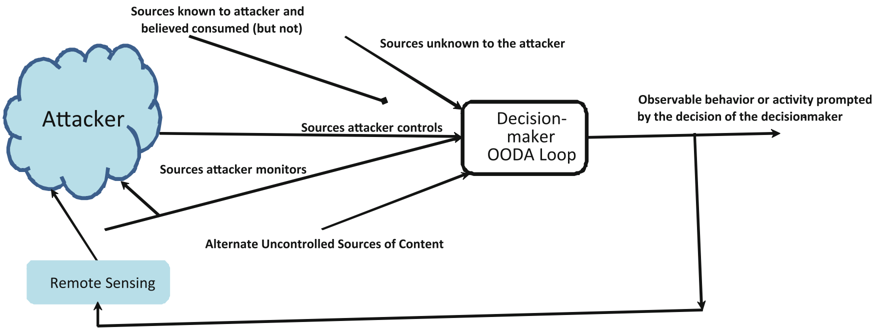


Fig. 9.3 Initial elemental model in a contested information environment

product of the decision. This is not to say the model of the decision-maker has a complete understanding of the datum content, but only an understanding derived by the decision-making model. The decision-maker in this elemental model may be a human decision-maker or an automated component. Obviously, the resources to model these two sources in a high-fidelity manner could be extremely high, and the level of fidelity to be achieved must be matched to the ultimate use of the modeling and simulation capability. As in any modeling endeavor, some level of abstraction must be adopted to make the model realizable. However, the decisions regarding abstraction must be made with recognition of the purpose of the model in mind. To this end, the following figures and paragraphs develop the model of human or automated decision-maker/problem-solver in a contested information environment. The initial very elemental model is shown in Fig. 9.3.

9.11 Sources and Content

The content being supplied by each of the sources is data. This model characteristic is not typical. It is typically assumed that the input to decision-making block is information. However, as we think about the content and its various formats in cyberspace, it is not unusual to think of the physical content being data packets, made up of combinations of ones and zeros, whose presence on a network of sources and consumers is arbitrated by hardware. It is presumed, using the *data in and activity out* type of model that understanding, and the transformation of data into information, is done within the decision-maker, be it a human or a component of automation. The fact that a decision could be made implies that some level of understanding based upon the data was reached before a decision could be made.

The National Institute of Standards and Technology published a guide to the measurement of information for security (Chew et al. 2008). Within that guide, the characteristics of availability, integrity, and confidentiality are described as

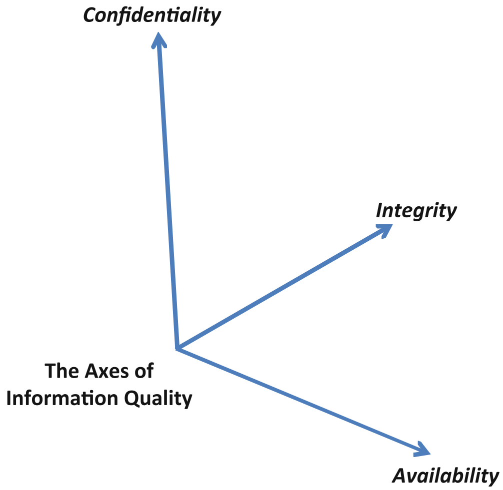


Fig. 9.4 Information quality from NIST SP 800-55 (Chew et al. 2008)

forming a comprehensive mapping of data quality. Of course, each of these characteristics is a combination of lowerlevel attributes that can be "rolled-up" into the higherlevel characteristic. If a quantification can be established for each of these three characteristics, and independence of the characteristics can be assumed, these characteristics could be viewed as an orthogonal three space. Thus, information quality can be thought of as a three space with an information source being located dynamically within the information quality three space. A figure depicting the information quality Three space is shown below as Fig. 9.4.

9.12 Models of the Decision-maker

The major target for information warfare offensive operations is the mind of an adversary, and specifically the problem-solving and decision-making activities of those minds. As stated previously, John Boyd's OODA loop is specifically called out in Joint Publication 3–13 as a model of the decision-maker to be targeted as well as the friendly decision-maker's OODA loop defended. This is visualized in the figure below in Fig. 9.5.

There are, however, many other models in the literature that may illuminate the decision-making process to a greater extent when it is operating in a contested information environment.

The models of decision-making to be found in the literature are numerous but typically fall into two categories, rational decision-making and naturalistic decision-making. Rational decision-making models, sometimes also called analytical models or logical models, utilize clear alternatives based on reliable data and

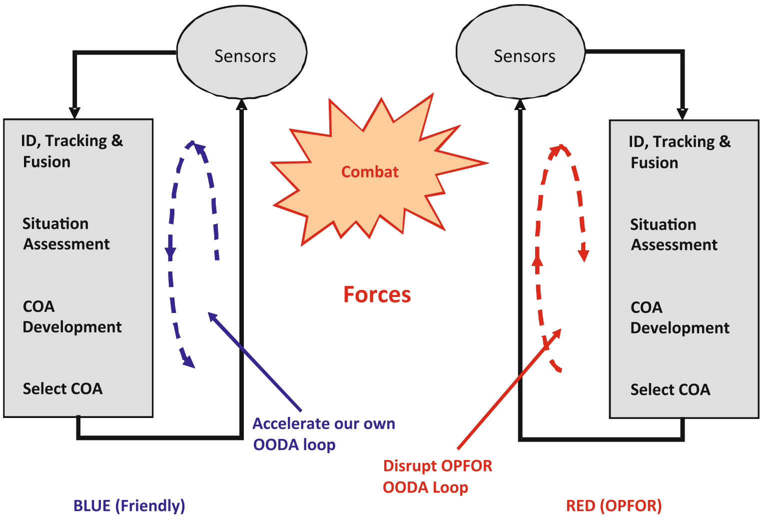


Fig. 9.5 Friendly and adversary OODA loops operating simultaneously

model environments in which there is typically a high level of confidence in the predicted response of the environment to the decision and resultant action. Naturalistic decision-making models, sometimes called action-based models, intuition models, or recognition-primed models, tend to link decisions to previously gathered knowledge and experience. They typically utilize decision refinement within the model.

Endsley and Jones (1997) reviewed decision-making models for their applicability to information warfare. In this review, they considered the general categories of decision-making models in the context of information warfare, or combat environment. They first describe "normative" models, sometimes called rational decision-maker models, and dismiss them as not being of much use for time-critical and dynamic situations (Endsley and Jones 1997). While this is likely true for a great number of decision-makers in the information warfare environment, it may not be true for several high-level decision-makers in the planning cycle and associated with the Joint Air Operations Center. This planning cycle typically operates on a 24 h cycle, with many of the strategic-level decisions being under consideration for a long period of time (relative to many tactical-level decisions which must be made while data are missing or uncertain). Endsley and Jones (1997) mention "long-range planning" as a possible exception, but do not appear to be talking about planners in the 24 h operational cycle. Endsley and Jones (1997) state in their review that naturalistic decision models are more likely to be utilized in the information warfare environment. They cite other researchers who determined by observation in the environment, that 95% of decisions were made through recognition of situational classification involving either situation matching (85%) or story-building (13%) (Endsley and Jones 1997). They also notie that building and maintaining situation awareness was critical to the decision-making process and was the major factor in determining the quality of the decision-making (Endsley and Jones 1997).

Azuma et al. (2006) recently reviewed models of time-critical decision-making. They reviewed several high-level models of military decision-making. Examples of these included the kill-chain model (composed of the elements of find, fix, track, target, engage, and assess), Lawson's model of command and control which parallels a feedback control loop, iterating towards a solution, and Wohl's SHOR model which Azuma states was inspired by behavioral mechanisms (Azuma et al. 2006).

Azuma et al. (2006) goes on in this review and describes the recognition/metacognition decision model (R/M). The R/M model is shown in Fig. 9.6.

Azuma et al. (2006) asserts that the R/M model is a dynamic and iterative problem-solving strategy in which, as the model operates, the next step is determined by the results of earlier steps, rather than a "global optimization" represented by the rational method. Azuma et al. (2006) states that it incrementally generates new hypotheses, tests, and goals and reconciles pattern recognition with problem-solving strategies, combining both use of experience to deal with routine decisions while also having an approach for handling uncertainty and novelty.

He states that the R/M model appears to be one of the most advanced models of time-critical decision-making and is an example of *adaptive expertise,* where an expert has deep domain knowledge but flexibility on his decision processes and structures to analyze and determine when they work and when they do not (Azuma et al. 2006).

From these articles, it appears that the R/M model is a naturalistic model appropriate for use in modeling the effect of information attacks in and through cyberspace on military decision-makers or any decision-maker in a time-critical environment.

9.13 Situation Awareness

While not strictly a model of decision-making, Endsley's model of Situation Awareness, or SA, envelops a decision-making element and in fact, is meant to model the processes utilized in making effective decisions. Endsley's model of SA is shown in Fig. 9.7.

Endsley's model of SA is composed of Individual Factors, shown at the bottom of the figure, the Task/System factors, shown in the middle of the figure, and the external factors, that could be thought of as the inputs to the task of the individual which would include decision-making. In the Endsley (1998) model, the information environment is contained inside the middle layer of the figure, depicted as the environment. The building and maintaining of situation awareness is performed at three levels, operating simultaneously, in the individual. Level 1 is perception, level 2 is comprehension, and level 3, the highest level, is projection

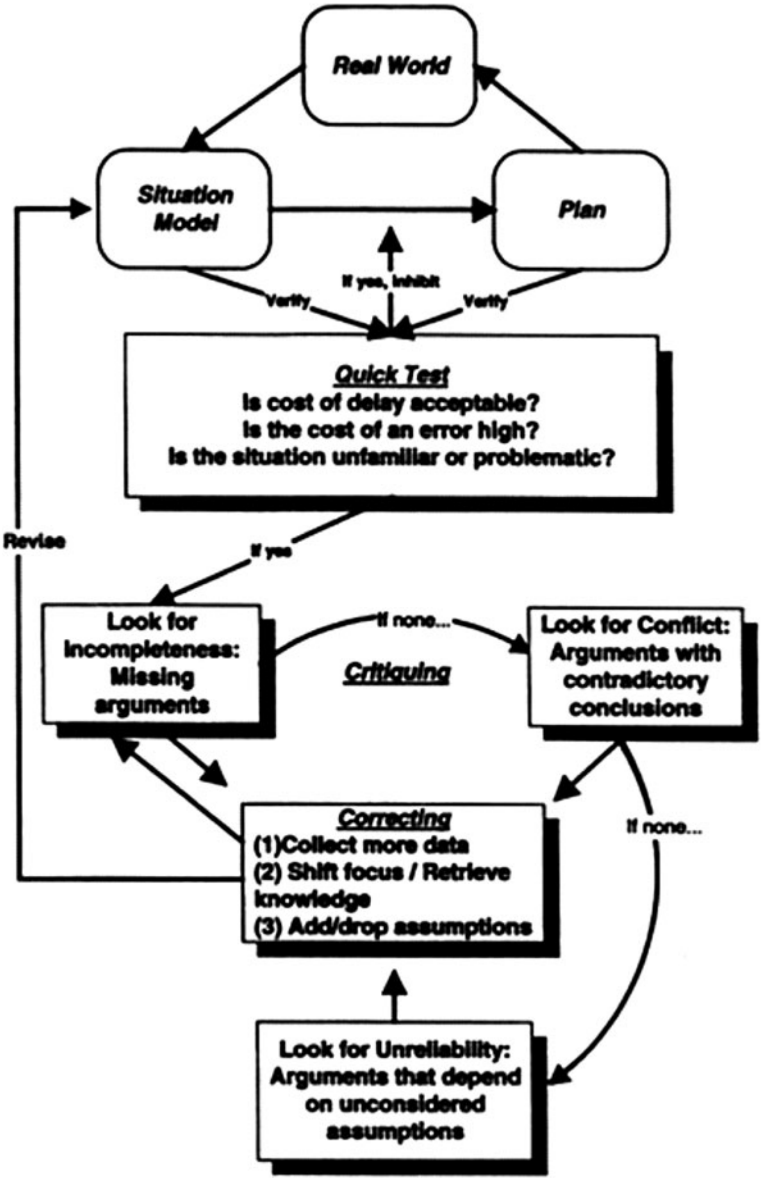


Fig. 9.6 R/M decision-making model (Azuma et al. 2006)

(or forward thinking). A much more thorough description of Endsley's model is available (Endsley 1998).

It appears from the literature that both the R/M and Endsley models may be appropriate for guiding the development of simulations to forecast the effect of information attacks in and through cyberspace on military operations.

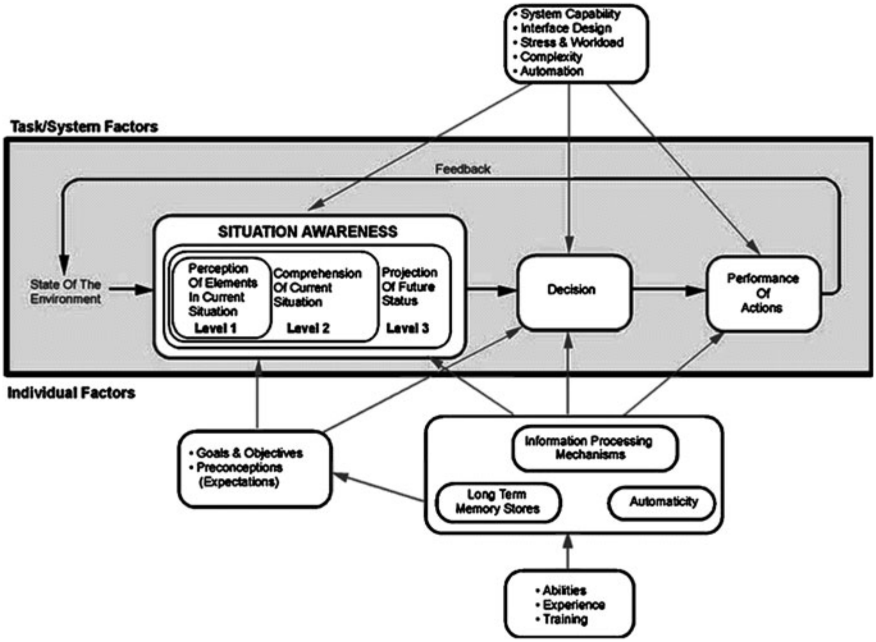


Fig. 9.7 Endsley's model of situation awareness (Endsley 1998)

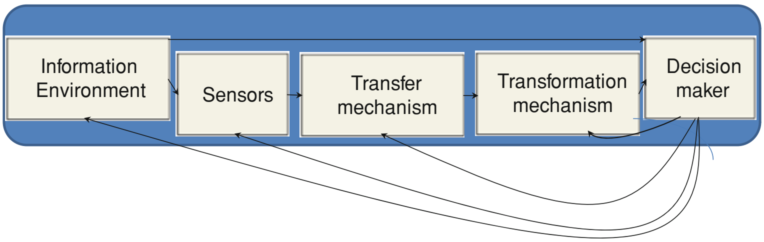


Fig. 9.8 Simulation basis for single decision-maker

9.14 Simulating the Contested Information Environment

Utilizing Fig. 9.3 as a basis, another high-level representation of a single decision-maker embedded in an information environment can be found in Fig. 9.8.

Figure 9.8 differs from Fig. 9.3 by making the flow from the information environment to the decision-maker visible. As in Fig. 9.3, the decision-maker may be a human that makes decisions or it may be automated capable of making decisions. Specifically, there must be a sensor, a transfer mechanism, and a transformational mechanism, between the information environment and the

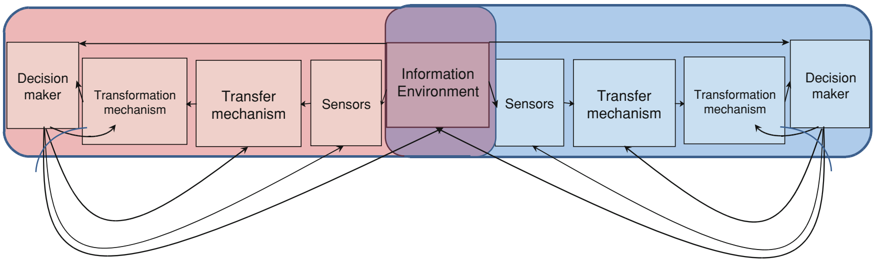


Fig. 9.9 Attacker and decision-maker contesting for the information environment

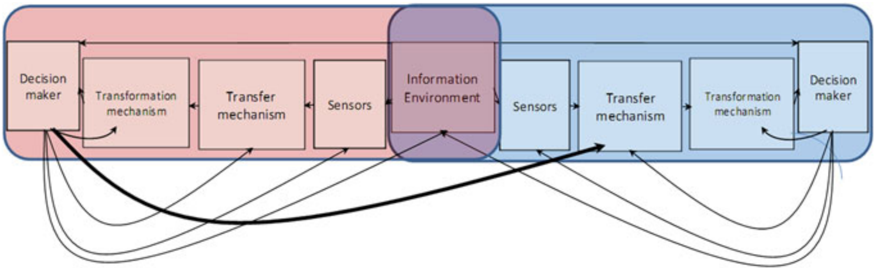


Fig. 9.10 Indicating an attack vector

decision-maker. It is assumed that the sensor will transform the data from the information environment based on the physics of the sensor itself and will transform that data a second time making it compatible to the transfer mechanism. As an example, the sensor may be an uninhabited aerial vehicle with an RF downlink, the transfer mechanism may be an RF receiver linked to a computer network implementing Ethernet and TCP/IP digital protocols, and the transformation mechanism may be a genera- purpose computer with an LCD-based visual display. There is also a direct path between the decision-maker and the environment. For a human decision-maker, this path could represent one or more of the human senses, such as a visual or auditory channel. As in Fig. 9.3, there is a path from the decision-maker to the information environment representing the product of decision-making having an effect on the environment. For simplicity, the transfer and transformation aspects of this channel are not depicted but do exist and may need to be modeled. The lines running from the decision-maker to the sensor, transfer mechanism, and transformation mechanism, indicates control over that block by the decision-maker. Figure 9.3 also includes a reference to an "attacker", representing some intelligence, again either human or automation, that controls part of the information environment, monitors part of the information environment that cannot be controlled, and is unknowing of other parts of the information environment. This addition is shown in Fig. 9.9.

In Fig. 9.9, the attacker is located to the left of the figure, having the same necessary components with which to interact in the Information Environment. It becomes clear from this figure that the attacker has several vectors with which an information attack could be conducted. These attack vectors can be shown on this figure as a line drawn from the red-side to a blue component, indicating control or an effect being applied to that component. An example of this is shown in Fig. 9.10 using the thickened black arrow to the blue transfer mechanism.

Not only could the attacker manipulate the information environment directly, but could also affect the sensor, the transfer mechanism, or the transformation mechanism of the decision-maker. These manipulations could be accomplished in series, in parallel, or in a specific temporal pattern. The attacker may also be able to monitor not only the information environment, but also the sensor, transfer mechanism, or the transformation mechanism of the decision-maker. Additionally, monitoring the decision-maker directly can also be indicated in this model. An example of such a direct monitoring capability may be a remotely controlled web camera located in the vicinity of the decision-maker.

9.15 Implementation of the Model in a Computational Simulation Environment

Joint Publication 3–13 identifies the three components of the information environment as being physical, information, and cognitive. These three components are shown in Fig. 9.11.

Information in this case refers to the content of various data transmission and storage technologies. These three components must be simulated in an integrated fashion to illuminate the effect information attack in and through cyberspace will have on military operations. There are several simulation developments described in the literature which have brought these elements together to simulate information operations over a military mission.

One such development was captured in the article by Paulhamus referenced earlier in this chapter. In this article, Paulhamus et al. (2009) analyzed the effect of information attack on the air and missile defense performance. In this article, a five step process was utilized to analyze the effect of information operations on a military capability, in this case air and missile defense. To perform this analysis, they evaluated several combinations of information operation type and multiple-unit force-on-force engagement-level air and missile defense scenarios using Monte-Carlo simulation. For their analysis, they integrated a limited set of information attack capabilities into the ACES (Burke MJ Henly 2002) air and missile defense simulation. This combination gave them the necessary validity with limited overhead (Paulhamus et al. 2009). While this approach is similar to what is needed, it does not offer the richness required for a mission environment nor does it offer a general framework for multiple and varied information attack vectors.

Davis (1995), in an extremely comprehensive discussion of modeling and simulation utilization in military contexts, describes distributed interactive

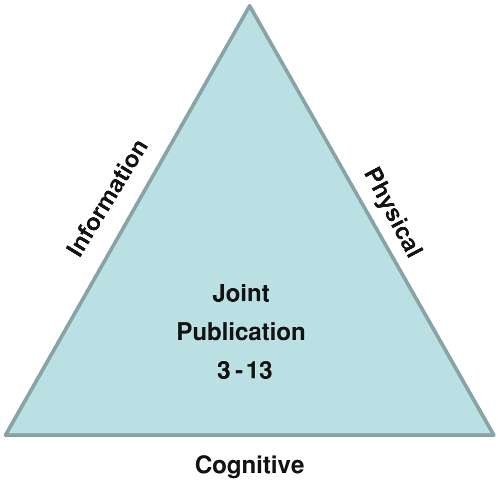


Fig. 9.11 The three components of the information environment

simulation (DIS) technologies as having the potential to be used to mediate planning and execution decisions for military missions. In this article, Davis describes the potential use of DIS in this way:

> ….DIS could be a powerful means for improving the quality of planning and analysis if it is used for occasional well designed experiments, sometimes experiential in nature and sometimes more systematic, to provide insights, to inform and calibrate models, and to test plans in a quasi-operational environment. Third, the emerging DIS culture may lower the artificial boundaries among developers, warfighters, and analysts, even if large-scale DIS operations are merely occasional. The challenges include developing appropriately integrated hierarchies of models, developing realistically adaptive decision models and other models of human behavior, developing and using new forms of uncertainty-sensitive analysis concerned more with exploration than finding definitive solutions, and learning when and how best to use DIS experiments (Davis 1995).

Dewar et al. (1996) developed, in a Rand report, a framework in which the use of a DIS-based simulation capability could be evaluated. In the vernacular of the report, the use of a DIS-based simulation to illuminate the effect of information attacks in and through, cyberspace during the planning and execution of military missions, would be to use it as an “analytical aid” and require it to be only “weakly predictive” (Dewar et al. 1996). Weakly predictive means, in this context that the simulation would be using several un-resolvable uncertainties as input, as would be expected when the input to the simulation may have been provided by intelligence sources in a time of conflict. The report further states that a DIS-based simulation would be useful if used under these conditions assuming that the results of the simulation, which may have the look of a highly prescriptive future, can be utilized appropriately (Dewar et al. 1996).

One of the shortcomings of the DIS-based simulation approach is that the cognitive aspects of the decision-making entities within the simulation are typically rule based and appear relatively crude and unintelligent, when confronted with an unusual problem set of the lack of information. This shortcoming was overcome by integrating computational cognitive models with the DIS-based simulation. One example of this is found in a 2005 thesis by Alford and Dudas which describes re-integrating TacAir SOAR into the DIS-simulation JSAF in an effort to validate the integration of the same cognitive model in a second simulation EEAGLES. The Dudas and Alford research developed a methodology to compare the behavior of the cognitive model, of a wingman flying a simulated fighter, as hosted within an earlier-developed simulation against the behavior exhibited by that same cognitive model hosted into a different simulation. In the methodology, if the hosting into the different simulation were to be judged successful, the behavior of the cognitive model must be the same in both host configurations. In this thesis, it proved difficult to re-host the cognitive agent into the DIS-based simulation due to changes in configuration of the DIS-based simulation that had been made by the developer since the cognitive model was last hosted, over 3 years prior (Alford and Dudas 2005). However, even when fully re-hosted and evaluated in aerobatic maneuvers using human-in-the-loop testing procedures, the cognitive model was not making the cognitive decisions and resultant observable behaviors that a lead pilot would have expected of a human wingman (Alford and Dudas 2005). In essence, the DIS-based simulation appeared to be a good platform in which to conduct a human-in-the-loop evaluation of a cognitive model's decision-making capability, supported by offline experimental data analyses. However, when contemplating utilizing this particular cognitive model for human-in-the-loop evaluation in a contested information environment, it appears it would be a very large software development effort, and subsequent validation effort, to modify and re-host the model before an evaluation could be conducted.

More recently, multi-agent system simulations customized for military warfare modeling have been utilized to evaluate effects of information manipulations on the performance of simulated military units. In 1994 Arthur, an economist, first described using multiple agent behavior to describe inductive reasoning. According to Arthur (1994), multi-agent systems have the ability to model human decision-making behavior that is not able to be modeled using more traditional models of rational decision-making. In general, this ability is based on multi-agent system's inherent ability to represent, and utilize, subjective belief structures in the decision-making process (Arthur 1994). This sentiment was echoed in 2002 by Cares when he advocated the use of agent-based models to support the development life cycle of a military concept. Cares (2002) segmented the development life cycle of a military concept into the following five steps and advocated the use of agent-based models, in differing configurations, during all five steps:

- Concept exploration
- Concept validation
- Deliberate analyzes
- War game adjudication and player support
- Field experimentation and operator support

In 2000 Naval Postgraduate School thesis, Roddy and Dickson referenced the Arthur work and used these multi-agent concepts to model human and organizational behavior and integrating agent behavior into an existing DIS-JAVA-VRML simulation environment of the kid's game *capture the flag.* Roddy and Dickson also noted that in 2000, there were several multi-agent simulation architectures in existence and briefly reviewed the capability of each. A thesis by Honabarger describes in 2006 using the SEAS multi-agent simulation to model network centric warfare constructs and then uses the model to evaluate metrics in military worth analyses . In his thesis, Honabarger utilizes a scenario representing the conflict in Kosovo, which was developed and validated in a previously published thesis, as a baseline case. Metrics taken from the baseline case represent warfighting performance in an information environment that is non-degraded. Degraded information environments are modeled in the Honabarger thesis by "turning-on" weather and terrain effects in geographicallybounded areas involved in the scenario that cause sensor performance and communication channel capacity reductions (Honabarger 2006). A Monte-Carlo experimental approach was utilized in this thesis. The description of the simulated degradation effects on sensor performance and communication channel capacity appear to be primarily availability reduction in the information environment and thus represent a single dimension in the three-space information environment graphical depiction in Fig. 9.11.

Honabarger (2006) indicates statistically significant effects observable in metrics representing each of the three dimentsions. This appears to confirm that a manipulation in a single dimension of the information environment may generate observable effects in other information environment dimensions.

It is also clear that DIS-based simulation capability and multi-agent simulation technologies bring differing, and uniquely useful characteristics to the modeling of the effect of cyber attack on mission operations. The multi-agent simulation capability robustly models naturalistic decision-making, which represents the cognitive dimension of the information environment depicted in Fig. 9.11, and is something DIS-based simulations have difficulty in accurately modeling with non-agent-based models. Alternatively, DIS-based models robustly model physical-dimension properties and real-time interactions for human-in-the-loop cognitive evaluations, depicted in Fig. 9.11, for which multi-agent simulations are not readily architected. Both simulation capabilities accurately model the content of the information environment, the "information" dimension depicted in Fig. 9.11. It would appear that the work conducted by Roddy and Dickson at the Naval Postgraduate school may be the best approach, and the best technological match, for modeling the information resource mapping to mission operations for the purpose of exploring the effect that cyber attack vectors may have on dynamic mission operations and the initial decision-making responses to those attacks using human-in-the-loop simulation techniques.

9.16 Conclusion

It is clear that the operations of government, industry, the military, academia, and even personal activity can be negatively affected by information attacks on and through cyberspace. This chapter has explored modeling constructs associated with understanding how changes in the information environment can affect decision-makers and problem-solvers attempting to maintain operations. Also in this chapter, how future modeling and simulation capability might be integrated into military planning and operational cycles was explored.

The military has taken several significant actions to increase and focus its efforts to defend against attacks in and through cyberspace while integrating its offensive kinetic and non-kinetic war fighting capabilities. However, the ability to know that the military can fight-through an attack in and through cyberspace before the attack takes place is still a research and development topic. Modeling and simulation may be capable of increasing the understanding of the potential affect of cyber attacks and guide the development of contingency planning. The increased understanding and more comprehensive and focused contingency planning will enhance the ability of organizations to assure their operation or mission, when operating in a contested information environment. This enhanced mission assurance will, increase the overall national security and will increase deterrence against the use of information attacks in the future.

Psychological experimentation and modeling have illuminated the cognitive constructs that are required to be included in future simulation efforts of conflicts in the information environment. Modeling and simulation efforts of the past focusing on conflicts within the information environment have helped to illuminate the way forward for future modeling and simulation efforts.

Future modeling and simulation efforts will aid organizations in their ability to understand the possible impacts to the organization's operations of cyber-based attacks to information availability, integrity, and confidentiality. These same modeling and simulation efforts will aid organizations to adaptively defend against the cyber-based threat, and compensate for degradations in the information environment during future attacks.

References

Air Force Doctrine Document 3–12 (2010) Cyberspace operations, 15 July 2010

Alberts D, Garstka J, Hayes R, Signori D (2001) Understanding information age warfare. Command and Control Research Program, CCRP 2001

Alford LE, Dudas BA (2005) Developing a validation methodology for TacAir SOAR agents in EAAGLES. M.S. thesis, September. Air Force Institute of Technology, Wright Patterson AFB

Arthur WB (1994) Inductive reasoning and bounded rationality (The El Farol Problem). Am Econ Rev (Pap and Proc) 84:406–411

Azuma R, Daily M, Furmanski C (2006) A review of time critical decision making models and human cognitive processes, 2006 IEEE aerospace conference, July 2006, pp 1–9

Beeker K, Mills RF, Grimaila MR (2010) Applying deterrence in cyberspace. IO J 1(4):21–27

Boyd J (1987) Slide series: a discourse on winning and losing, Maxwell air force base. Tech Rep M-U 43947, August

Burke MJ, Henly JM (2002) The APL coordinated engagement simulation (ACES). John Hopkins APL Technical Digest 23(1)

Cares J (2002) The use of agent-based models in military concept development, Winter simulation conference

Castelli CJ (2008) Defense department adopts new definition of cyberspace. Inside the air force (23 May 2008). http://integrator.hanscom.af.mil/2008/May/05292008/05292008-24.htm

Chew E, Swanson M, Stine K, Bartol N, Brown A, Robinson W (2008) NIST Special Publication 800-55-Revision 1: Performance measurement guide for information security, National Institute of Standards and Technology, Gaithersburg, MD, July

Davis PK (1995) Distributed interactive simulation in the evolution of DoD warfare modeling and simulation. Proc IEEE 83(8):1138–1155

Dewar JA, Bankes SC, Hodges JS, Lucas T, Saunders-Newton DK, Vye P (1996) Credible uses of the distributed interactive simulation (DIS). Rand Corp, Santa Monica, CA ADA306986

Endsley MR (1998) Toward a theory of situation awareness in dynamic systems. Hum Factors 37(1):32–64

Endsley MR, Jones WM (1997) Situation awareness, information dominance, and information warfare. Technical Report AL/CF-TR-1997-0156, Armstrong Laboratory, WPAFB,OH 45433, February

Fortson LW and Grimaila MR (2007) Development of a defensive cyber damage assessment framework. In: Proceedings of the 2007 international conference on information warfare and security (ICIW 2007), Naval Postgraduate School, Monterey

Fry SA (2009) Joint Publication 1-02: Department of defense dictionary of military and associated terms, Joint Chiefs of Staff, US Department of Defense. http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf

Grimaila MR (2008) Improving the cyber incident mission impact assessment process. In: Proceedings of the cyber security and information intelligence research workshop (CSIIRW 2008), Oak Ridge National Laboratory, Oak Ridge

Grimaila MR, Fortson LW, Sutton JL and Mills RF (2009a) Developing methods for timely and relevant mission impact estimation. In: Proceedings of the 2009 SPIE defense, security and sensing conference (SPIE DSS 2009), Orlando, FL

Grimaila MR, Schechtman G, Mills RF (2009b) Improving cyber incident notification in military operations. In: Proceedings of the 2009 Insitute of industrial engineers annual conference (IERC 2009), Miami, FL

Grimaila MR, Fortson LW, Sutton JL (2009c) Design considerations for a cyber incident mission impact assessment (CIMIA) process. In: Proceedings of the 2009 international conference on security and management (SAM09), Las Vegas, Nevada

Hellesen D, Grimaila MR, Fortson LW, Mills RF (2008) Information asset value quantification. In: Proceedings of the 2008 international conference on information warfare and security (ICIW 2008), Peter Kiewit Institute, University of Nebraska Omaha, 24–25 April 2008

Honabarger JB (2006) Modeling network centric warfare (NCW) with the systems effectiveness ananlysis simulation (SEAS), M.S. thesis, Air Force Institute of Technology, WPAFB, OH

Joint Publication 3–13 (2006) Information operations, 13 February 2006

Joint Publication 3–30 (2010) Command and control for joint air operations, 12 January 2010

Libicki M (1996) What is information warfare? Center for Advanced Concepts and Technology, National Defense University

Libicki M (2007) Conquest in cyberspace: National security and information warfare. Cambridge University Press, New York

Libicki M (2009) Deterrence in cyberspace, High Frontier, May 2009. http://www.afspc.af.mil/shared/media/document/AFD-090519-102.pdf

Mauney C (2009) Speech, space weapons in the twentyfirst century. http://www.stratcom.mil/speeches/19/

Mullen MG (2009) Joint Publication 1: doctrine for the armed forces of the USA, Joint Chiefs of Staff, United States Department of Defense. http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf

Paul C (2008) Information operations: doctrine and practice: a reference handbook. Praeger Security International, Westport

Paulhamus BL, Castner AK, Jailall RRJ, Sibley NW, Snow KZ (2009) Analyzing the effect of information attack on air and missile defense performance. Military Communications Conference, MILCOM2009

Roddy KA, Dickson MR (2000) Modeling human and organizational behaviorusing a relational-centric multi-agent system design paradigm, Naval Postgraduate School, M.S. thesis, Monterey, CA, September

Sorrels D, Grimaila MR, Fortson LW, Mills RF (2008) An architecture for cyber incident mission impact assessment (CIMIA). In: Proceedings of the 2008 international conference on information warfare and security (ICIW 2008), Peter Kiewit Institute, University of Nebraska, Omaha

Tinnel LS, Saydjari OS, Haines JW (2003) An integrated cyber panel system. In: Proceedings of the 2003 DARPA information survivability conference and exposition, vol 2, pp 32–34

Tzu S (1994) The art of war (trans: Lionel Giles 1910). (Electronic text in the public domain May 1994). http://classics.mit.edu/Tzu/artwar.html

Ullman DG (2007) OO-OO-OO! The sound of a broken OODA loop, Crosstalk: J def Softw Engineer

United States Strategic Command (2006) Deterrence operations, Joint operating concept, version 2.0, Director, Plans and Policy, USSTRATCOM, Offutt AFB NE 68113. http://www.dtic.mil/futurejointwarfare

US Government Accountability Office (2006) Internet infrastructure-challenges in developing a public/private recovery plan, GAO-06-1100T, Accessed 13 Sept 2006

Verizon Business RISK Team (2009) Data breach investigations report. Verizon Business. http://www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf

Vijayan J (2005) Data security risks missing from disaster recovery plans. Comput World 39:41. Accessed 10 Oct 2005

Waltz E (1998) Information warfare principles and operations. Artech House, Norwood

Wood R (1995) Information engineering: the foundations of information warfare. Air War College, Maxwell AFB, GA

Chapter 10
Interactive Model-Based Decision Making for Time-Critical Vehicle Routing

Subhashini Ganapathy, Sasanka Prabhala, S. Narayanan, Raymond R. Hill and Jennie J. Gallimore

Abstract Advances in technology, software algorithms, and operations research methods provide the opportunity for effectively coupling the human decision maker with optimization modelling algorithms in large-scale systems operating in dynamic and uncertain environments. In military applications, such as search and rescue/destroy missions or real-time route planning or re-planning provide time windows within which critical decisions need to be made. Using a specially constructed human-computer integrated routing application, an evaluation was conducted to compare the effects of interactive model-based solutions with respect to automated solutions generated by mathematical modelling algorithms in the context of unmanned aerial vehicle route planning. Results indicate that significantly more high priority targets were covered in the human integrated approach compared to the

S. Ganapathy (✉)
User Experience Research Group, Intel Corporation,
5400 NE Elam Young Pkwy, Hillsboro, OR 97124, USA
e-mail: Subhashini.ganapathy@intel.com

S. Prabhala
Interactions and Experience Research Group, Intel Corporation,
2111 NE 25th Avenue, Hillsboro, OR 97124, USA
e-mail: Sasanka.v.prabhala@intel.com

S. Narayanan · J. J. Gallimore
Department of Biomedical, Industrial, and Human Factors Engineering,
Wright State University, 207 Russ Engineering Centre,
3640 Col. Glenn Hwy, Dayton, OH 45435, USA
e-mail: S.narayanan@wright.edu

J. J. Gallimore
e-mail: Jennie.gallimore@wright.edu

R. R. Hill
Department of Operational Sciences, Air Force Institute of Technology,
2950 Hobson Way, WPAFB, OH 45433, USA
e-mail: Raymond.hill@afit.edu

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_10,

automated solution without any significant degradation with respect to all the other dependent measures including percentage of total targets covered, low priority targets covered, total targets covered in threat zone, high priority targets covered in threat zone, and low priority targets covered in threat zone.

10.1 Introduction

Many real-world applications such as military planning problems, supply and logistics, vehicle routing, production planning, manufacturing, and health care planning systems require taking actions in time-critical situations. The command and control of these complex, dynamic systems often requires integrating human operators into the decision making process of the system. Successful system performance then depends on the effectiveness and efficiency of human–computer interactions and the timeliness of the resultant output. Delays and failures in making decisions in these applications, are often expensive in terms of money, system performance, and maybe even human lives. Hence, system designs based on incorrect models of decision aids may result in a human-automation system that is, in practice, less effective than the human-based non-automated system (Evans et al. 1991; Parasuraman et al. 1999).

Both humans and computer algorithms have strengths and limitations that can be brought into play in a joint cognitive problem solving approach. Human cognitive skills are limited when voluminous data must be interpreted and analyzed, whereas computer algorithms can fail when heuristic or intuitive knowledge about the system is required. A hybrid computer-aided and operator-aided solution can potentially improve the overall performance of the system. Previous studies have demonstrated that having humans-in-the-loop can enhance system performance (Ammons et al. 1998; Ruff et al. 2002; Li 2000). However, there are limited studies on systematic approaches to human-centred automation particularly involving the coupling of optimization methods and human reasoning in a joint problem solving process and task allocation among the human and computer algorithm, particularly in complex dynamic tasks.

Research involving interactive systems seems to vary on how to distribute the solution finding task among the human and computer algorithms. In some of the systems, the human interacts with the computer algorithm only in the initial stages of solution finding (Harder et al. 2004; Wang and Shen 1989). The human controls the input to the algorithm while the solution generation process remains essentially a black box. In other cases, the user is included only as an afterthought (Schneider et al. 2000). There are several issues associated with the interactive modelling of human-centred automation, such as trust in automation, passive role of human operator, out-of-the-loop performance, and opacity/transparency of automation. A proper understanding of the automated tools, underlying system algorithms, and user needs is critical to realize advanced human integrated system concepts. However, the collective knowledge base is still somewhat deficient in how to best allocate functions between the human and the automation components. Thus, there remains a pressing need for studies into and theories regarding the coordination between human operators and automated controllers in the remotely operated vehicle domain as well as systematic studies of human/system interface development in command and control of such vehicles.

We have developed an interactive model-based decision tool that systematically couples human intuitive knowledge and optimization methods involving unmanned aerial vehicles (UAVs) in a simulated vehicle route planning/replanning scenario for target identification missions. Our vehicle routing scenario provides a rich foundation for modelling human-centred decision making as it involves a system that is complex, dynamic, and uncertain. The remainder of the article presents an overview of related research on human-centred automation, details the implementation of the simulation architecture, and discusses the results of empirical evaluation comparing interactive model-based solutions to solutions generated purely based on mathematical model and algorithms.

10.1.1 Levels of Automation

Automation means full or partial replacement of human functions within some system of intent. The extent of automation varies across a range from no automation to full automation. Various levels of automation have been classified in different studies (Endsley and Kaber 1987). There are problems cited when implementing the extremes, a fully automated system or a fully manual system. In the case of a fully automated system, research literature indicates that automation may fail to improve the performance of the system due to various reasons such as (a) oversimplification of the model, (b) not responding at the right time due to lack of intuitive knowledge, (c) automation biases (Mosier et al. 1998; Smith et al. 1997), and (d) out-of-the-loop performance problems (Barnes and Matz 1998; Endsley and Kiris 1995; Entin et al. 1995; Thackray and Touchstone 1989).

There are various issues that can degrade performance in a fully manual system. This is due to the dynamic and complex nature of system data handled by the human operators in planning, decision making, and executing. First, human operators tend to narrow their attention with respect to the task (i.e., if more tasks are present, the amount of attention given to each task is narrowed to accommodate all the tasks). Second, human operators tend to focus on the dominant factors. This is known as cognitive tunnel vision (Sheridan 1997). Third, the human operator can make decisions too early, before exploring all the possibilities; Endsley (1996) termed this premature closure. Fourth, it takes a relatively long time for human operators to retrieve information from their long-term memory. This could pose a problem, especially in real-time situations, when the human operator must make decisions under time pressures.

There is evidence that systems perform better when humans and machines operate in combination (information gathering, information analyzing, and decision making) versus when the system is operated in either fully manual (humans alone) or in fully automated mode (machine alone) (Prabhala and Gallimore 2004; Ruff et al. 2002). A study conducted by Jentsch and Bowers (1996) showed that automation manipulations improved over all task performance only when used in a combination of humans and machines and not when each was used separately.

The human must be involved with automation in a meaningful way to overcome the problems associated with automation. Realizing the limitations of the extreme levels of automation, a number of interactive modelling methods have been proposed for multi-criterion decision making. The next section discusses the body of literature in interactive modelling and the limitations and assumptions of previous studies.

10.2 Interactive Modelling

Realizing the limitations of traditional methods, there are a number of interactive methods proposed for multi-criterion decision making that attempt to address some of the automation issues mentioned earlier. These interactive methods rely on decision maker preference information generated interactively during the optimization. Interactive modelling potentially augments the strengths of humans in complex decision making, such as providing correct information for better visual perception (Scott et al. 2002), including strategic assessment, and accommodating uncertainty handling. Interactive, human-centred, model-based approaches help in managing complexity and provide useful insights into the features required for the support of human problem solving and decision making tasks to potentially improve the performance of the complex system through the joint human-machine system. Li and Li (2010) in their paper talk about the advantage of a hybrid system along with simulation models to help effective decision making.

There is anecdotal evidence that interactive systems are better than completely manual or completely automated systems. Related research efforts of interactive systems are summarized in Table 10.1. The first column references the study. The second column describes the domain for which the interactive optimization was used. The third column lists the interactive optimization mechanism. The fourth column lists some of the assumptions/limitations of the study. The fifth column lists the interaction modalities and extent of human interaction for the given domain, and the last column lists the optimization algorithm applied in the context. These studies provide specific instances of interactive systems but there remains a lack of a systematic study of human-centred modelling approaches for these systems. This paper investigates the research issues in effective joint cognitive problem solving for a class of problems related to vehicle routing.

Table 10.1 Overview of interactive modelling research efforts

Reference	Domain	Mechanism	Assumptions/limitations	Level of human interaction	Underlying optimization technique
Klau et al. (2002)	Graph layout problem, delivery; Protein; Job shop scheduling	Algorithm presents a solution and the user can accept, reject or invoke a different solution	Single solution is presented to the user Each problem contains finite number of elements (deterministic) Solution alteration is based on a single parameter	Visual Metaphors Modify solution by modifying single parameter	Tabu Search
Venugopal and Narendra (1990)	Multiple objective optimization-general purpose MODM problems	Interactive optimization	Decision maker will be able to provide information, when required, during the course of interaction Feasible region defined by the constrained set is convex Supports deterministic system	Operator can improve upon the solution by specifying the set of objectives that need to be improved	Nash bargaining principle
Waters (1984)	Vehicle routing	Interactive simulation	Humans specify parameters before a solution is generated Not interactive while achieving the solution	Computer algorithm is as a note-pad and calculator User interacts using a menu-driven program	–
Harder et al. (2004)	Vehicle routing	Parameter tweaking	Humans specify parameters initially Multiple solutions can be presented	Parameter setting	Tabu search

(continued)

Table 10.1 (continued)

Reference	Domain	Mechanism	Assumptions/limitations	Level of human interaction	Underlying optimization technique
Nulty and Ratliff (1991)	Fleet scheduling	Interactive optimization	Solution specific to the problem	Specifies parameters Tunes the solution	Relaxed integer program
Schneider et al. (2000)	Logistics planning	Post optimization	Must be able to interact with the optimization algorithm Human–computer interaction is not iterative	Perform what–ifs on the result set	Genetic algorithm
Fisher (1985)	Scheduling	Parameter tweaking; Post optimization	Modified results are not recomputed Supports only trained or expert users Human–computer interaction is not iterative	Using graphics Assigning customers thus modifying the results	Bargaining principle

10.3 Domain Description

Vehicle routing applications span a wide variety of applications including commercial distribution of products, dial-a-ride, street sweeping, and military applications in routing of combat vehicles. Vehicle routing is defined as the problem of determining a best set of routes for pickup or delivery of supplies to different locations or customers in a distributed system. The vehicle routing problem (VRP) is an important component of many logistics and distribution management systems. Dantzig and Ramser (1959) first described and mathematically formulated VRPs. Since then, research has examined different aspects of the VRP. Significant reviews of routing problems include (Bodin et al. 1983; Golden and Assad 1988 Laporte 1992; Ombuki et al. 2006; Cordeau et al. 2002).

The primary objective of most routing problems is to minimize the total cost of providing service. This cost could include the vehicle resource costs, mileage, or personnel costs. For emergency services, such as ambulance, police, or fire services, minimizing response time to an incident may be a primary objective. Other objectives include: (a) minimization of the transportation cost, (b) minimizing the number of vehicles used, and (c) minimizing the penalties associated with partial service of the customers.

Typical constraints associated with VRPs include: (a) vehicle capacity, (b) travel time, (c) assignment of certain number of vehicles to certain customers, (d) driving regulations such as working period during the day, maximum duration of driving period, and overtime, (e) operational constraints such as the nature of the goods transported, perishable or non-perishable; (f) time windows for customer delivery, (g) precedence constraints, such as collection and then delivery of material; (h) backhaul constraints associated with loading and unloading operations, (i) road constraints such as one-way street, no left turn, etc.; and (j) grouping or sequencing of customers (Toth and Vigo 2002).

The domain we investigate is the routing of UAVs for time-critical target identification under hostile situations. UAVs have been widely used in the areas of military intelligence surveillance and reconnaissance. The UAVs have been operational in Bosnia and used to monitor buildings, military forces, and battle activities in support of NATO. It is quite possible that future UAV operations will involve the surveillance and location of terrorist activities and training facilities. The specific scenario used in this study was adapted from the notional set of Bosnia reconnaissance targets used in (O' Rourke et al. 2001). In our scenario the human operator supervises a set of UAVs and is responsible for rerouting those UAVs when pop-up targets are identified and assigned. The overall goal of the planning mission is to route the vehicles to cover the maximum number of targets based on factors such as priorities of the targets, restricted fly zones, and the loiter time of the UAVs. Each target is associated with a low priority or a high priority. The location of the targets can determine whether the target is present in a restricted fly zone such as a threat zone. Loiter time is the service time that the vehicle spends on covering a target.

The decision to reroute the UAVs is based on the perspective of decision makers involved in the decision. The principle factors used in forming the perspective of the decision maker are target coverage, target priorities, and/or restricted fly zones. The human operator selects a route based on their perspective and what they are principally looking for in a routing solution. The weights associated with the principle factors are dynamic and may change with respect to the operator's current assessment of the scenario and the data presented through any real-time information feeds. The human integrated approach provides an iterative approach that lets the solution evolve and be improved by the human operator with their inputs, without having to reformulate the underlying routing problem.

For our scenario the routing of any vehicle to targets is based on two techniques—an automated method and a human interaction-based method. In the automated method, the underlying optimization algorithm is based on Tabu Search. Tabu Search is a metaheuristic technique used to generate sub-optimal but generally good solutions (Harder et al. 2004). A distinguishing feature of the Tabu Search is its exploitation of adaptive forms of search memory allowing the search to cover a wider region of the overall search space. The Tabu Search performs a responsive exploration for alternative solutions. This responsive exploration is based on the fundamental assumption that a strategic search yields more useful information than a purely random search. This information can be exploited to create even more efficient search processes. When a new pop-up target is assigned to the operator a new automated route is calculated based on Tabu Search algorithm using target priorities. The algorithm for the generation of the automated route is based on the assumption that pop-up targets are covered regardless of priority, as all targets pose a potential threat in the military domain. The new route is graphically presented to the human operator and includes information such as targets covered by the vehicles, targets not covered by the vehicles, the order of the targets covered, and the loiter time associated with the UAVs near each target.

In the human integrated solution method, the human operator can interactively modify the current solution to generate a new solution. This new solution combines both the human knowledge and the Tabu Search based optimization algorithm. This human interaction method is potentially valuable when the operator perceives an improvement on the automated solution based on their mental model of the various solution criteria and their domain specific experience. Since the scenario is a multi-criteria problem, with the goal to achieve several objectives at once, and since this may not be done with a single solution, the human operator has the flexibility to iteratively modify a solution, thereby generating a number of solutions giving the operator choices for a solution. The processes adopted by the human, in solution generation, can be comprehensively represented using the cognitive modelling method outlined in the next section.

10.4 Supervisory Control Model

In a supervisory control mode, humans are involved in cognitive functions such as problem solving, judgment, decision making, attention, perception and memory (Norman 1986; Sheridan 1997). Cognitive engineering techniques such as operator function model (OFM) and task analyzes, model domain tasks in terms of the goals of operators and the methods available to operators to achieve those goals. The cognitive load on the human operators or supervisory controllers is driven by the continual need for situation assessment, active goal-setting and planning, and anticipatory as well as reactive control actions and compensating for abnormal system conditions (Jones and Mitchell 1994).

The OFM can be used both predictively and descriptively to explain the operator action, or in some cases, lack of occurrence of an operator action (Mitchell 1987). In order to develop the cognitive model of the operator, it is essential to understand the tasks performed by the operator and the content and form of information that should be presented to the operator. These models can be used to support the design, implementation, and evaluation of interactive systems. In order to develop the human operator model we applied OFM representation techniques.

The OFM is widely used to model human action by providing a mathematical and visual representation of operator activities in the control of complex, dynamic systems. It is structured both heterachically and hierarchically to model the operator tasks. In an OFM, the nodes represent the action and the arcs describe the events that lead to the operator activities.

The primary function of the operator, in our system, is to reroute vehicles in order to cover any pop-up targets. In order to reroute the vehicles the operator must analyze the target location and priority, and then review the solution in order to maximize the targets covered. The next step is to select the solution and reroute the vehicles. Figure 10.1 represents the OFM of supervisory control of vehicle routing. The top level functions include (a) re-routing the vehicles for the mission, (b) analyzing the vehicles and target locations, priorities for re-routing, (c) analyzing the solution, and (d) selecting a solution for re-routing the vehicles.

The selection of the solution is decomposed into three sub-tasks based on whether the operator would choose the automated solution generation approach or interactive solution approach. These are denoted by arcs 1 and 2, respectively. The possible states of decision making under the human integrated solution assessment are as follows:

1. Avoid threat zones in the path of the vehicles when selecting targets;
2. Select targets that need to be covered by the vehicles;
3. Specify loiter time for the vehicle at the specified target; and
4. Assign a vehicle to cover a target and recalculate the solution.

The recalculate solution can be reached from one of the three possible states – select targets, specify loiter time, assign vehicle. These are denoted by arcs 3–5 in

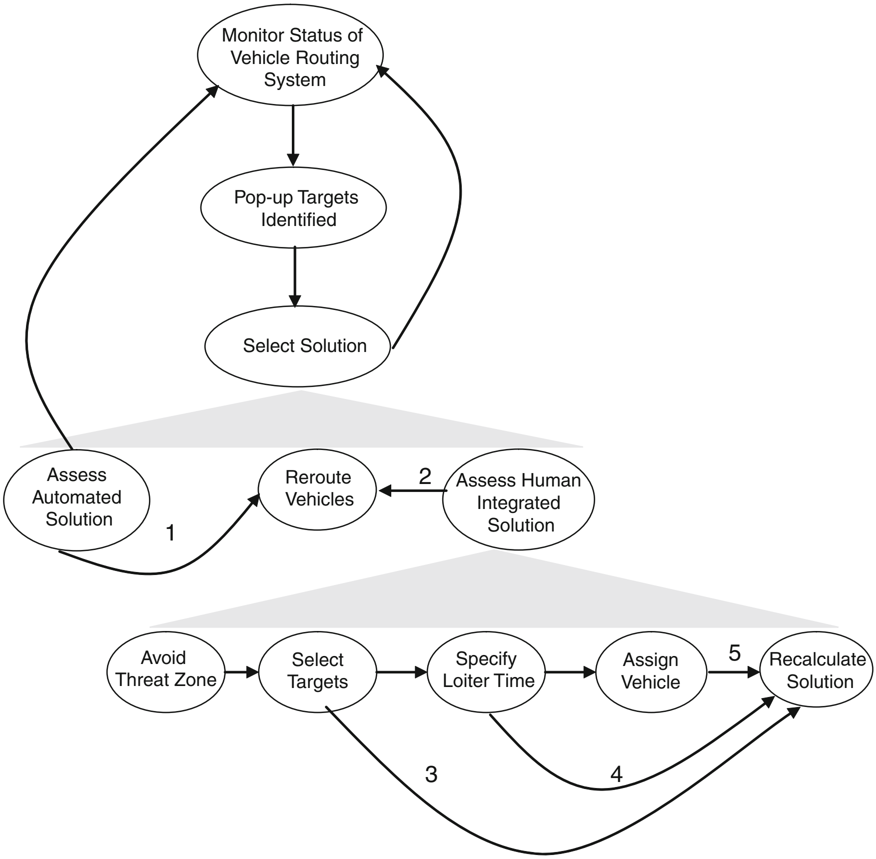


Fig. 10.1 An OFM of a supervisory controller in UAV routing

Fig. 10.1. This modelling method was utilized in developing a model-based user interface for the supervisory controller responsible for a simulated reconnaissance mission.

10.5 Scenario Description

To assess the effectiveness of the model-based approach, we simulated a reconnaissance mission using a Java® Programming language-based simulation embedded within an extension of the AFIT Router (Harder et al. 2004). Figure 10.2 displays a snapshot of the operator console for the simulated mission. The operator console interface allows operators to monitor and control multiple UAVs travelling along various waypoints in order to identify targets. At the start of the simulation, the operator is presented with two UAVs each travelling along predetermined flight paths. The interface displays the map of the Bosnia scenario described earlier. Each UAV starts from a base location and moves along a route of waypoints visiting assigned targets. The route is represented by a line connecting waypoints. The targets are represented by blue squares. The red label below the target location indicates the target priority as high and the black label represents the target priority as low. At a specified time '*t*', the simulation pauses and pop-up targets appear on the screen. The pop-up targets are represented as green triangles on the interface. Pop-up targets are time-critical targets tasked to the operator (and the vehicle) by higher headquarters. Pop-up targets require deviation from currently executing routes to accommodate coverage of the target followed by resumption of the prior route consisting of the older targets. Optimally routing among remaining targets once a pop-up target is serviced is not a trivial task.

When pop-up targets appear, the human operator is responsible for re-routing the vehicles in order to accommodate the new targets. As shown in the upper right corner of the interface, an automated solution is presented to the operator. The new route is represented by bold lines and the completed route to date represented by dashed lines. The operator can identify the target name, the order, and loiter time by moving the cursor over the target.

The operator can accept the automated solution, analyze the automated solution, or reject the automated solution. If the operator accepts or rejects the automated solution, the appropriate solution will be selected and the vehicles will follow the selected path. The operator may reject the automated solution, if the operator feels more high priority targets are left uncovered in the automated solution or if the operator feels they can improve the automated solution presented by using their knowledge about the system, past experiences, and/or criticality of the mission as inputs.

If the operator chooses to analyze the automated solution, an interactive panel is presented. The interactive panel provides system information such as the targets skipped in the automated solution (represented with a rectangle around the label), loiter time associated with targets, location of the targets (threat zone or not), and the vehicle assignment to the targets. The priority of the targets is also represented in the label colour of the targets. Red represents high priority targets and black represents low priority targets. Based on their heuristics or intuitive knowledge, the operator can vary loiter time, assign a specific vehicle to a target, or select the targets that need to be covered. This information is then sent back to the optimization module and the new route is presented to the operator, as shown in the lower right corner of Fig. 10.2. The operator can then accept or reject the new interactive solution. The time remaining to make a decision is presented to the user at the top right corner of the interface. The time remaining is determined based on the state change of the UAVs. Within this time, the operator can iteratively interact with the computer algorithm and can generate solutions if the operator perceives an improvement in the solution presented. The simulation architecture was developed to facilitate both fully automated solution generation and hybrid human-aided computer-aided solution generation.

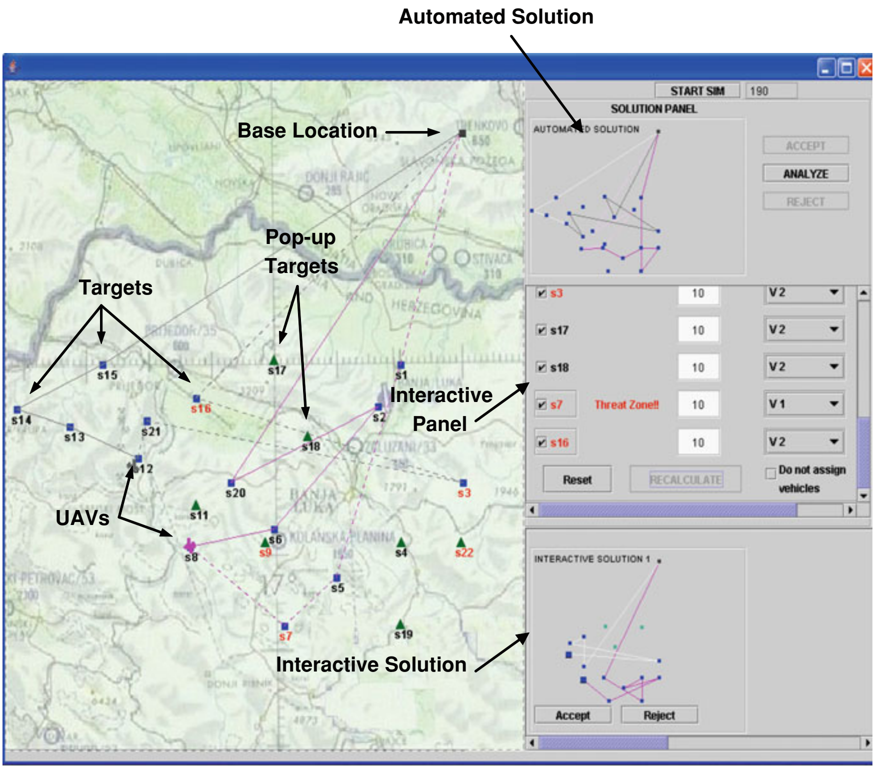


Fig. 10.2 Snapshot of the UAV routing mission simulation

10.6 Methodology

This study evaluates the effect of model-based decision making in improving the efficiency of the vehicle routing system. The objective of this study was to determine whether the human integrated approach lead to better solution generation.

10.6.1 Independent and Dependent Variables

The simulation was used to investigate the effect of an independent variable (type of solution generated) while monitoring and routing UAVs in a time-critical system. The two levels of "type of solution generated" investigated were: Automated solution and Human interactive solution. To evaluate the efficiency of the solution type generated, seven dependent variables were analyzed:

- *Percentage of total targets covered*: The total targets visited or covered divided by the total number of targets to be covered in a trial.
- *Percentage of high priority targets covered*: The high priority targets covered divided by the total number of high priority targets to be covered in a trial.
- *Percentage of low priority targets covered*: The low priority targets covered divided by the total number of low priority targets to be covered in a trial.
- *Percentage of total targets covered in threat zone*: The total targets covered in threat zone divided by the total number of targets to be covered in threat zone in a trial.
- *Percentage of high priority targets covered in threat zone*: The high priority targets covered in threat zone divided by the total number of high priority targets to be covered in threat zone in a trial.
- *Percentage of low priority targets covered in threat zone*: The low priority targets covered in threat zone divided by the total number of low priority targets to be covered in threat zone in a trial.
- *Number of times interactive solution was used*: Number of times the interactive solution was used to successfully route the UAVs to cover the targets in a trial.

10.6.2 Participants

Twelve graduate student volunteers from Wright State University participated in this experiment. All participants were screened to have normal or corrected 20/20 vision and colour vision capabilities. This criterion was important as participants should be able to differentiate between different types of targets (high priority, low priority, and pop-up targets), different UAVs, and also between the new and completed UAV routes.

10.6.3 Apparatus

The simulation was written in Java® and run on a 3.20 GHz personal computing system running Windows XP. A 17-inch LCD monitor was used to display the interface, with a mouse and keyboard used as the input devices. The experiment took place in an office type environment with dim lighting. The participants sat in an adjustable office chair, and the mouse and keyboard were placed at a comfortable position as determined by each participant.

10.6.4 Procedure

Participants were trained to use the interface for UAV monitoring and routing tasks. Specific tasks taught during training included: detecting pop-up targets,

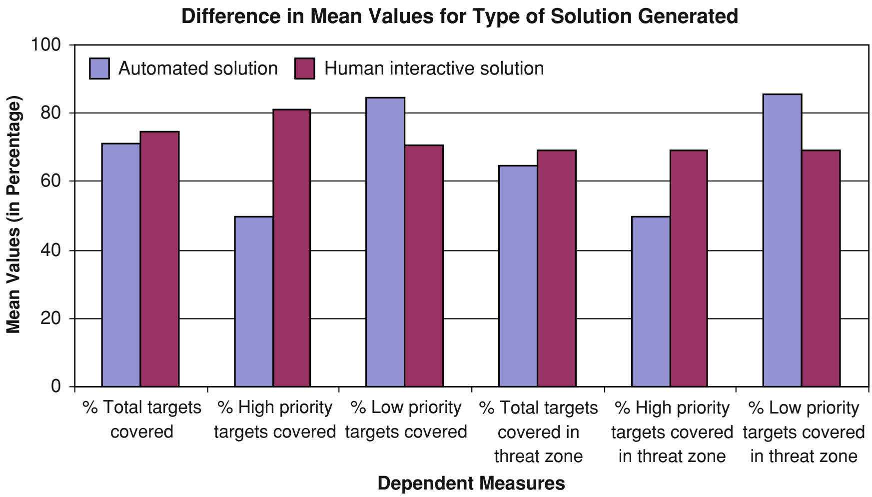


Fig. 10.3 Difference in mean values for type of solution generated

identifying high and low priority targets, assigning UAVs to pop-up targets, assigning UAVs to high and low priority targets, identifying threat zones, and to monitor mission time and progress. Participants were not timed and took as long as necessary with the training until they were comfortable with the interface. Upon completion of training, participants ran five experimental trials. The total number of targets covered, location of the targets, and the base location of the UAVs were varied in each experimental trial to avoid participants learning effects.

During any given trial, participants monitored two UAVs travelling around specified paths to cover the targets. At a specified time (different for each of the experimental trial) the simulation paused and pop-up targets appeared on the interface, an automated solution was also displayed on the interface. The pop-up targets appeared in random locations. The participant could accept, reject, or analyze the automated solution. If the analyze button was selected the participant would have to re-route the UAVs. Responses include mouse clicks and keyboard data entry. Each experimental trial lasted 7 min. Time was displayed for the participants in seconds from ‘420’ counting down to ‘0’. At the end of each trial, the participants were asked to rank order the following factors considered for the generation of the solution: priority of targets covered, total number of targets covered, and the total time remaining for solution generation. In addition, the number of times the participants used the human integrated solution approach to achieve a solution was also collected.

10.7 Results

A parametric analysis of the dependent measures was conducted to test for statistically significant difference in the type of solution generated. A two-tailed t-test was conducted for each of the dependent measures. Results indicated that there was a significant difference for the dependent measure percentage of high priority targets covered t (0.025, 58) = 2.3152, ($p = 0.0242$). A significantly greater percentage of high priority targets are covered ($X = 81.193$) in human interactive solution than in the automated solution ($X = 50.0$). There were no other statistically significant differences.

While statistically significant differences were not found for most of the dependent variables there is a trend for improved performance with respect to the human interactive solution. Figure 10.3 illustrates the differences in the mean values between the automated solution and the human interactive solution by for each of the dependent measures. The percentage of total targets covered in a human interactive solution ($X = 74.778$) was slightly higher than in an automated solution ($X = 71.154$), and within this total a statistically significant greater number of high priority targets were covered using the human interactive solution ($X = 81.193$) compared to the automated solution ($X = 50.0$) as previously indicated. The automated solution covered 13.65% more low priority targets. When targets were in threat zone, the difference between the two conditions was only 4.48% (human interactive solution $X = 69.189$ and automated solution $X = 64.706$). However when targets were in the threat zone a greater percentage of high priority targets were covered in the human interactive solution condition ($X = 68.966$) compared to automated solution ($X = 50.0$).

Participants using the human integrated solution ranked three factors on which they based their solution. The rank ordering of the factors is illustrated in Table 10.2 highlighted in bold. Sixty nine percent of the participants ranked the priority of targets as their first consideration. Total number of targets was ranked second 56.36% of the time, and time remaining for solution generation was ranked third, 74.54% of the time. The average number of times participants used the human integrated approach was found to be 1.66.

10.8 Discussion

The primary tenet of an integrated human–machine system is to increase the effectiveness of decision makers in situations where the computer can support and enhance human judgment in the decision making. This paper investigated the effectiveness of a model-based approach to enable humans' to interactively generate solutions in a VRP. A preliminary evaluation was conducted to compare the effects of interactive model-based solutions with respect to automated solutions generated purely based on a mathematical model and algorithms. Results indicated

Table 10.2 Ranking of participants' decision factor

Factors	Rank 1	Rank 2	Rank 3	Not ranked
Priority of targets	**69.09**	20	5.45	5.45
Total targets covered	29.09	**56.36**	12.72	1.82
Time remaining	5.45	12.72	**74.54**	7.27

that a significantly higher number of high priority targets were covered in the human integrated approach compared to the automated solution.

The automated solution showed a trend for covering a higher percentage of low priority targets when targets were in the threat zone. While not statistically significant this outcome can affect the performance of the system in terms of the effective use of resources such as ammunitions and fuel used in visiting the low priority targets. Moreover, in the case of suppression of enemy air defence (SEAD) mission and search and rescue/destroy missions destroying a low priority target rather than a high priority target may affect the nature of the mission and the outcome of the war. Considering these possible cost outcomes, the differences of approximately 19% fewer higher priority targets covered in the threat zone under automated condition is very meaningful. Based on the ranking and results it is obvious that participants considered priority as the most important factor.

Lack of statistically significant differences may be based on ceiling effects for the dependent variable total targets covered. The percentage of total targets that could be covered in the specified time frame is set based on the mission time. Additionally, the small subject pool may have resulted in low statistical power. It is suggested that a meaningful measurement of cost for missing high priority targets to cover low priority targets be created and analyzed in future research.

This study found that humans use their ability to detect subtle changes in environments, intuitive knowledge, and apply knowledge based on past experiences to present situations to achieve the goal or objective. The average number of times participants used the human integrated approach and participant rankings indicated that the participants did not use a 'trial-and-error' technique to come up with the solution. Instead, they used their knowledge about the system as the input to the solution generator and came up with good solutions using the human integrated solution approach.

Traditional models (Batez et al. 1990; Massaglia and Ostanello 1989) do not use human input for generation of solutions. The model solution is generated purely based on mathematical model and algorithms. Such approaches may fail when applied to complex systems due to unforeseen events or the dynamic nature of interactions.

Our proposed approach enables human operators to concurrently evaluate multiple feasible alternatives. During this process, they gain insights on the solution being evaluated and its impact on system performance. This coupling of the human to the solution process could help alleviate the problems associated with human computer interactions such as opacity, situational awareness, and human error.

This research contributes to the body of knowledge in interactive optimization by focusing on effective approaches to combine human capabilities and optimization algorithms in the context of VRPs. The initial results are encouraging as better solutions were generated without a significant loss on performance. The use of modelling and simulation tools has helped in defining multiple solutions in a quick and easy fashion. Future research will focus on systematically validating the results using operational experts and expanding the scope of an integrated system.

Acknowledgments S. Narayanan, R. Hill, and S. Ganapathy were supported in part by Air Force Office of Scientific Research grant FA9550-04-1-0163.

References

Ammons JC, Govindaraj T, Mitchell CM (1998) Decision models for aiding FMS scheduling and control. IEEE Trans Syst Man Cybern 18(5):744–756

Barnes MJ, Matz MF (1998) Crew simulation for unmanned aerial vehicle (UAV) applications: sustained effects, shift factors, interface issues, and crew size. In Proceedings of the human factors and ergonomics society 42nd annual meeting, pp 143–147

Batez BW, Pas EI, Neebe AW (1990) Generating alternative solutions for dynamic programming based planning problem. Socio Econ Plan Sci 24(1):27–34

Bodin LD, Golden BL, Assad AA, Ball M (1983) The state of the art in the routing and scheduling of vehicles and crews. Comput Oper Res 10(2):163–212

Cordeau JF, Gendreau M, Lapotr G, Potvin JY, Semet F (2002) A guide to vehicle routing heuristics. J Oper Res Soc 53(5):512–522

Dantzig GB, Ramser RH (1959) The truck dispatching problem. Manag Sci 6:80–91

Endsley MR (1996) Automation and situational awareness. In: Parasuraman R, Mouloua M (eds) Automation and human performance: theory and applications, pp 163–181

Endsley MR, Kaber DB (1987) Level of automation effects on performance, situation awareness and workload in a dynamic control task. Ergonomics 42(3):462–492

Endsley MR, Kiris EO (1995) The out-of-the-loop performance problem and level of control in automation. Hum Factors 37(2):381–394

Entin EB, Entin EE, Millan J, Serfaty D (1995) Situational awareness and human performance in target recognition. In: IEEE international conference on systems, man and cybernetics. pp 3833–3837

Evans GW, Stuckman B, Mollaghasemi M (1991) Multicriteria optimization of simulation models. In: Winter simulation conference, pp 894–900

Fisher ML (1985) Interactive optimization. Ann Oper Res 5:541–556

Golden BL, Assad AA (1988) Vehicle routing: methods and studies. In: Golden BL, Assad AA (eds) Studies in management science and systems. Elsevier, NY

Harder WR, Hill RR, Moore JT (2004) A Java^{TM} universal vehicle router for routing unmanned aerial vehicle. Int Trans Oper Res 11:259–275

Jentsch F, Bowers C (1996) Automation and crew performance: the importance of who and what. In Proceedings of the human factors and ergonomics society 40th annual meeting, pp 46–53

Jones PM, Mitchell CM (1994) Model-based communicative acts: human-computer collaboration in supervisory control. Int J Hum-Comput Stud 41:527–551

Klau GW, Lesh NB, Marks JW, Mitzenmacher M (2002) Human-guided tabu search. In Proceedings of national conference on artificial intelligence (AAAI), pp 41–47

Laporte G (1992) The vehicle routing problem: an overview of exact and approximate algorithms. Eur J Oper Res 59:345–358

Li S (2000) The development of a hybrid intelligent system for developing marketing strategy. Decis Support Syst 27(4):395–409

Li S, Li JZ (2010) Agents international: integration of multiple agents, simulation, knowledge bases and fuzzy logic for international marketing decision making, expert systems with applications. Int J 37(3):2580–2587

Massaglia and Ostanello (1989) N-tomic: a support system for multicriteria segmentation problems. In: Korhone P, Lewandowski A, Wallenious J (eds) Multiple criteria decision support. Springer-Verlag, Berlin, pp 167–174

Mitchell CM (1987) GT-MSOCC: a domain for research on human-computer interaction and decision aiding in supervisory control systems. IEEE Trans Syst Man Cybern 17(4):553–572

Mosier KL, Skitka LJ, Heers S, Burdick M (1998) Automation bias: decision making and performance in high-tech cockpits. Int J Aviat Psychol 8(1):47–63

Norman DA (1986) Cognitive engineering. In: Norman DA, Draper SW (eds) User centered system design: new perspectives on human computer interaction. Lawrence Erlbaum Associates, NJ, pp 31–61

Nulty WG, Ratliff HD (1991) Interactive optimization methodology for fleet scheduling. Nav Res Logist 38:669–677

O' Rourke KP, Bailey TG, Hill RR, Carlton WB (2001) Dynamic routing of unmanned aerial vehicles using reactive tabu search. Mil Oper Res 6(1):5–30

Ombuki B, Ross BJ, Hanshar F (2006) Multi-objective genetic algorithms for vehicle routing problem with time windows. Appl Intell 24(1):17–30

Parasuraman R, Mouloua M, Hilburn H (1999) Adaptive aiding and adaptive task allocation enhance human–machine interaction. In: Scerbo M, Mouloua M (eds) Automated technology and human performance: current research and trend. Lawrence Erlbaum Associates, NJ, pp 119−123

Prabhala S, Gallimore JJ (2004) Investigation of error rates when controlling multiple uninhabited combat aerial vehicles. In Proceedings of the winter simulation conference, pp 1026–1031

Ruff HA, Narayanan S, Draper M (2002) Human interaction with levels of automation and decision aid fidelity in the supervisory control of multiple simulated teleoperated air vehicles. Presence Teleoper Virtual Environ 11(4):335–351

Schneider NL, Narayanan S, Patel C (2000) Integrating genetic algorithms and interactive simulations for airbase logistics planning. In: Suzuki Y, Roy R, Ovaska S, Furuhashi T, Dote Y (eds) Soft computing in industrial applications. Springer-Verlag, NY, pp 309–318

Scott SD, Lesh N, Klau WG (2002) Investigating human-computer optimization. In Proceedings of CHI conference, pp 155–162

Sheridan TB (1997) Supervisory control. In: Salvendy G (ed) Handbook of human factors and ergonomics. John Wiley & Sons, NJ, pp 1295–1327

Smith PJ, McCoy CE, Layton C (1997) Brittleness in the design of cooperative problem-solving systems: the effects on user performance. IEEE Trans Syst Man Cybern 27(3):360–371

Thackray RI, Touchstone RM (1989) Detection efficiency on an air traffic control monitoring task with and without computer aiding. Aviat Space Environ Med 60(8):744–748

Toth P, Vigo D (2002) The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia

Venugopal V, Narendra TT (1990) An interactive procedure for multiobjective optimization using Nash bargaining principle. Decis Support Syst 6(3):261–268

Wang HF, Shen HF (1989) Group decision support with MOLP applications. IEEE Trans Syst Man Cybern 19(1):143–153

Waters CDJ (1984) Interactive vehicle routing. J Oper Res Society 35(9):821–826

Chapter 11
PerFECT: An Automated Framework for Training on the Fly

Hari Thiruvengada, Anand Tharanathan and Paul Derby

Abstract Currently available cognitive training systems can highly benefit from more adaptable and encapsulated frameworks that include better performance assessment methods, robust feedback mechanisms and automated mechanisms that reduce the manual intervention and curriculum management required during training sessions. In short, there is an ardent need for an automated human in the loop training system that can effectively train cognitive skills required for military operations. An automated training system would be extremely beneficial if it can be easily coupled with a synthetic learning environment to function autonomously is an entirely data driven manner. Such a system would enable rapid deployment of key training scenarios, skills and tactics to war fighters and help them maintain a superior level of competence in the battlefield. An automated framework for training on the fly also known as performance feedback engine for conflict training (PerFECT) which includes key components for simulating training scenarios, measuring trainee's performance, providing relevant feedback and dynamic curriculum management is discussed in this chapter. First, the training system comprises of custom plug-in interface that allows components of the training framework to readily interface with a simulated virtual learning environment. Second, it has a "Performance Evaluator" that enables automated, real-time and objective evaluation of a trainee's performance grounded within an objective framework known as time window and enables run-time evaluation of performance skills based on a skills matrix. Third, PerFECT has a "Feedback System" that can provide contextual and immediate feedback to trainees based on process measures. Finally, PerFECT includes a "Curriculum Manager" that dynamically selects appropriate training scenario from a template library with varying levels of complexity. The selection algorithm for training scenario is based on the trainee's

H. Thiruvengada (✉) · A. Tharanathan · P. Derby
Honeywell ACS Labs, 1985 Douglas Drive N, Minneapolis, MN 55442, USA
e-mail: hari.thiruvengada@honeywell.com

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_11,

historical performance scores and complexity of the earlier scenarios. We also present the initial findings from a pilot study which helps illustrate the capabilities of the framework and conclude with future directions in this area of research.

11.1 Introduction

For decades, researchers have been building complex and advanced simulator-based training systems that present challenging training scenarios, support objective performance evaluation, provide unambiguous feedback, and adjust the training curriculum to adapt to the changing tactics within the scenario. However, these components are often constructed independently and cannot be easily interfaced with each other. Furthermore, each of these components requires some level of manual or supervisory interaction from the human counterpart. Prior to constructing these automated training systems one must consider the integration needs (Barnett 2009) in simulated virtual environments and address several key issues (Parasuraman and Riley 1997) related to automation of human machine training systems. First and foremost, there is often a common disconnect that occurs in the way people process information and automated systems process information both in real world and virtual or simulated environments. This phenomenon is referred to as "automation surprise" (Sarter and Woods 1997). Second, automated systems often tend to negatively impact vigilance (Billings 1997; Grier et al. 2003), which is important to sustain good performance over time. This phenomenon also referred to as "vigilance decrement" (Grubb et al. 1995) can cause boredom and/or workload fatigue at the wrong time. Third, when a failure occurs in a complex automated system, it usually presents a significant challenge to the user as they may not be able to recover from the failure due to the lack of understanding of how the system handled the failure. This situation might even worsen as the human may not notice the failure immediately and have trouble switching to manual mode (Sarter and Woods 1997). Fourth, humans often place trust in automated systems which is equivalent to the level of trust they place on people in real world. Due to this excessive trust placed on automated systems, they often tend to become overconfident in the ability of the automated system to complete the procedural tasks at hand which leads to a phenomenon known as "automation bias" (Mosier et al. 1997). When such a failure occurs, people tend to lose confidence in the system and they never place the same level of confidence in the automation system again (Lee and Moray 1992; Moray et al. 2000). Finally, humans possess and share a vast amount of implicit knowledge (Barnett 2009) that is often not understood or shared by automated systems. As a result, humans tend to perform better in situations that require the use of such tacit yet critical knowledge.

11.2 Requirements Analysis

This section highlights the keys requirements of automated training systems and provides more details on the components that are required to realize such a system.

11.2.1 Key Requirements

In traditional training exercises, a retired or active-duty senior non-commissioned officer (NCO) acts as observer/controller (OC) during field training exercises (Foltz et al. 2009). After developing the training scenarios, the OC manages the execution of the training mission including deploying opposing forces (OPFOR) to exploit any observed vulnerabilities or errors. After completing the mission, the OC leads an after action review (AAR) to provide feedback on trainee performance. Currently this labor-intensive activity requires many weeks to develop training scenarios, travel to training facility, execute training missions, and receive critical feedback. In addition, traditional training methods present other significant challenges such as, slow adaptation to new tactics, high cost to develop and operate advanced training systems, constrained evaluation mechanisms in which experienced trainers observe trainees, incompetent feedback mechanisms, lack of adaptation to trainee performance and the incapability to observe and measure some of the trainee behaviors and decisions.

In short, following are some of the several critical limitations in the traditional, state-of-the-art training systems:

- *Manual intervention*: An experienced trainer is required to update latest in the field experiences. This is not done automatically.
- *Lack of adaptation*: Traditional training systems cannot keep up with the fast pace of changing threats and new tactics.
- *Difficulty in assessment*: Traditional training systems are often incapable of tracking, observing and measuring non-overt actions and trainee decisions.
- *Poor performance assessment methods*: Traditional training systems are based purely on overt trainee actions, and this is not sufficient to enhance learning in trainees.
- *Slow deployment*: With traditional training systems it is extremely difficult to deploy new best practices and scenarios based on current field reports within a short duration.

11.2.2 Skills Matrix: Mapping Skills to Tasks

The term “training” suggests that trainees systematically learn about specific competencies, such as knowledge, skills, and attitudes (KSA) that are required to successfully perform in a work environment. Similarly, the term “team” training suggests that a trainee (or a set of trainees) will systematically learn about the specific KSA that are required to successfully perform on a team within a work environment (Salas and Cannon-Bowers 1997).

Knowledge In its most elementary form, knowledge can be defined as "acquired information that can be activated in a timely fashion in order to generate an appropriate response" (Schultetus and Charness 1999). In other words, knowledge includes what people need to know to perform a task. This could include facts, rules, procedures, and strategies (Gordon 1994; Lewandowsky et al. 2007). For example, knowledge consists of a team's shared mental model, interpositional knowledge, teamwork knowledge, and team knowledge (Cannon-Bowers and Salas 1997). The ability for a team to share knowledge leads to better task performance and improves efficacy (Cannon-Bowers and Salas 2001).

Skills Skills are behaviors needed to perform a specific task. A few examples of individual skills include situation awareness (Endsley 1995), communication, coordination, etc. On the other hand, team skills include the team member's adaptability, shared situation awareness, metacognitive abilities, team management, coordination, communication, and decision making (Cannon-Bowers and Salas 1997)..

Attitudes Lastly, attitudes refer to how an individual or team members feel about the task or the team. Attitudes may comprise of collective efficacy, collective orientation, cohesion, and attitude toward teamwork (Cannon-Bowers and Salas 1997).

While developing an effective training program for battlefield operations, it is essential to be sensitive to the complexity and dynamic nature of the actual battlefield environment. War fighters should be trained to maintain the highest level of performance while adapting to complex environments that change continuously. Barnett (2009) points out that the critical road block in the development of automated systems is that humans process information differently than automated systems. While a software module within an automated system processes information discreetly using bits of zeros and ones, human processes information in a fuzzy manner. In other words, information is processed using fuzzy probabilistic networks that are made up of several nerves that are triggered by nerve impulses that may be synchronous or asynchronous. Another important distinction is human processes information based on pattern matching techniques whereas automated systems require exact matches to perceive information. While designing an automated training framework, these key differences must be considered to achieve maximum efficiency.

11.2.3 Virtual Environments as a Pedagogy Tool

Simulated learning environment (SLE) typically includes simulations, games, and virtual environments. SLEs are useful because they create and/or augment real experiences that people have in the real world (Cannon-Bowers et al. 2008).

In fact, most SLEs can recreate environments that are too dangerous or inaccessible in the real world (Mantovani 2001).

One type of the SLE is the virtual environment (VE) (Schmorrow et al. 2009; Nicholson et al. 2009; Cohn et al. 2009), which is a simulated, computer generated environment that creates "synthetic sensory experiences" (Salas et al. 2002). For example, users could interact with computer generated images, sounds, and haptics. Multiuser virtual environments (MUVEs), also referred to as virtual worlds (VW) and net-worked virtual environments, allow users to interact with other users within the simulated environment for the purpose of collaboration and knowledge sharing (Bainbridge 2007; Salas et al. 2002). Oftentimes, virtual words mimic complex physical environments and are inhabited by other users represented by animated characters, or avatars (Bainbridge 2007). Two examples of MUVEs used for training purposes are Second Life (Linden Labs) and OLIVE (Forterra Systems, now part of SAIC).

Virtual environments have proved to be successful and effective training environments (Anderson et al. 2001; van Dongen et al. 2007; Verdaasdonk et al. 2005) due to the following reasons. First, virtual environments allow trainees to develop cognitive skills and knowledge while being situated within an enriched high fidelity environment that mimics a real environment (Dieterle and Clarke 2008; Roman and Brown 2008). The enables then to maintain context while operating. Trainees are given the opportunity to think and act in the mode of the particular domain (Shaffer and Resnick 1999). Second, virtual environments are effective for training due to the flexibility they afford for experimentation (Bainbridge 2007; Lampton et al. 2002; Loomis et al. 1999; Riva 1997). It is possible to tailor a given virtual environment to a specific research question. For example, experimenters are able to control variables and measure responses and performance within the environment (Loomis et al. 1999; Riva 1997).

11.2.4 Automated Training Framework for Simulated Learning Environments

The event-based approach to training (EBAT) provides a framework for the entire lifecycle of training (Johnson et al. 1997; Salas and Cannon-Bowers 1997; Salas et al. 2006). This method has been adapted for a variety of settings (Oser et al. 1999), including training within virtual environments (Salas et al. 2002). EBAT provides a framework from the entire lifecycle of training. EBAT consist of the implementing the following steps: (1) skill assessment, (2) learning objectives, (3) scenario events, (4) performance measurements, (5) performance diagnosis, and (6) feedback. To achieve complete automation within the training system, the EBAT approach must be adapted to include performance mediated curriculum management. In order to be acceptable and effective, an automated training framework must cater to the needs of the key steps established within the EBAT framework.

Skill Assessment The Assessment. The first step of EBAT is to determine the team's level of knowledge, skills, and attitudes relevant to the training topic (Salas et al. 2002; Bewley et al. 2009). Training developers should elicit both the trainee's current KSA as well as an expert's KSAs. The trainee's KSAs are compared to the expert's KSAs in order to determine what KSAs are deficient. Typically, training program developers will extract this information using traditional and cognitive task analyzes (Kirwan and Ainsworth 1992; Salas and Cannon-Bowers 2001).

Learning Objectives Next, learning objectives are created in response to the deficiencies identified in the skill inventory. That is, the learning objectives are based on the discrepancies between the trainee KSAs and the expert KSAs. These learning objectives become the focus of the training, which become the basis for each scenario (Salas and Cannon-Bowers 1997; Salas et al. 2002, 2006). Each learning objective should be measureable by means of an objective score, a performance score, an embedded automated score, or an expert rating.

Scenario Events Based on the learning objectives, "trigger events" are created to test the competencies of the team members with respect to each learning objective (Salas and Cannon-Bowers 1997; Salas et al. 2002, 2006). There are several benefits to incorporating scenario events into virtual environments (Salas et al. 2002). First, virtual environments are able to adapt to the training needs of individual trainees. For example, a curriculum management system could be used to manage the order and complexity of the scenarios presented to individual trainees based on their previous performance. Second, trainees would be given the opportunity to repeatedly practice and master the variety of trigger events within a given scenario (Cannon-Bowers and Salas 2001). This type of practice is not always possible in group training events.

Performance Measures Performance measures are used to assess the user's mastery of the learning objectives. Each measure should be sensitive (i.e., able to assess difference in performance) and diagnostic (i.e., able to interpret performance; Meister 2004; Cannon-Bowers et al. 2007). For individual training, the trainer would want to assess both the processes (i.e., how the trainee accomplished a task) and the outcomes (i.e., whether or not the trainee accomplished a task) for each learning objective of the training scenario (Cannon-Bowers and Salas 1997). However, for a team, a trainer would need a more multifaceted approach. That is, the experimenter would want to assess the processes and outcomes of both the individual and the team (Cannon-Bowers and Salas 1997).

Performance Diagnosis Performance measures are interpreted in this stage of EBAT (Salas et al. 2002). This interpretation diagnoses the trainee's KSAs and identifies problems with the trainee's mental model. By targeting deficiencies in KSA, the experimenter is able to build constructive feedback related to the learning objectives. In addition, deficiencies are used as a basis for adjusting future training methods. That is, the learning objectives are reestablished and the training curriculum is arranged appropriately.

Feedback Feedback supplies the trainee with a focused description of what behavioral and cognitive changes need to be made in order to improve performance (Salas et al. 2002). Feedback is an essential mechanism for learning (Canon-Bowers et al. 1998). Feedback provides participants with knowledge about what they accomplished, how they accomplished it, how well they accomplished it, and what they could do to perform better. Depending on the nature of the training, the feedback mechanism can be provided in a variety of ways in terms of content and schedule.

There are several ways to provide feedback content to the trainees (van Buskirk et al. 2009). First, experimenters could issue experimental feedback, or information about the relationships between the cues within the environment and their respective outcomes (Balzer et al. 1994). Second, experimenters could give trainees normative feedback, or knowledge about how he or she is performing in relation to others (Smithers et al. 1995). Third, experimenters could simply provide outcome feedback, or knowledge about the results of the trainee's performance (Ericsson et al. 1993). Fourth, experimenters could provide process feedback, or how the trainee should or can perform the task (Kluger and DeNisi 1996). Lastly, experimenters could provide progress feedback, or knowledge about how the trainee's performance has progressed over time (Kozlowski et al. 2001).

Curriculum Management The last missing piece which is essential to enable a fully automated EBAT training framework is curriculum management. If the curriculum management component mediated by current and past performance and scenario complexity, then it would be helpful in adjusting the learning pace of the trainee.

11.3 Automated Training Framework

The performance feedback engine for conflict training (PerFECT) training framework addresses the key challenges by placing trainees in simulated learning environment (OLIVE) and exposing them to appropriate training scenarios with apt time compression to focus on critical skills; collect performance metrics based on cognitive, behavioral, environmental and human interaction models; provide relevant and timely feedback to the trainee; and automatically select additional novel training scenarios based on identified skill deficiencies. To aid the training on the fly, the performance evaluation mechanism uses a skills matrix that is based on skills, rules and knowledge model based metrics (Rasmussen 1983; Rasmussen 1986). The skills within the skills matrix were developed and mapped to specific tasks within the training scenarios.

11.3.1 System Architecture

To demonstrate the feasibility of our architecture, we implemented the PerFECT framework within the context of a military fire team that comprises of fire team

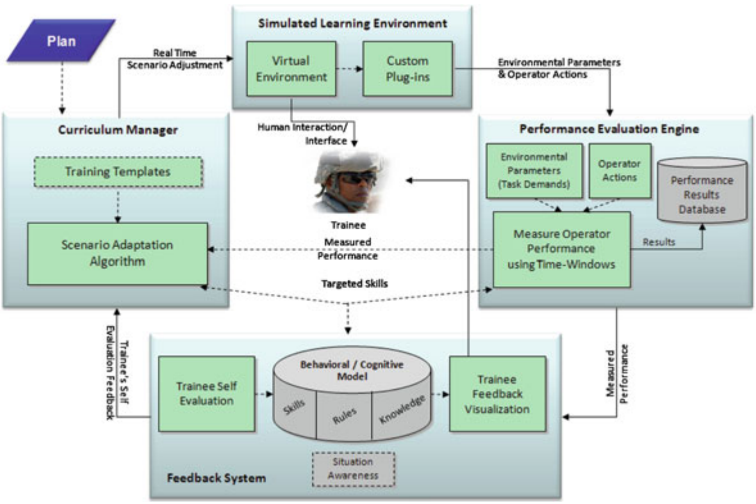


Fig. 11.1 Overall architecture of PerFECT framework

leader (FTL) and three fire team members (FTMs). The FTL is a good target for this training framework since they deal with significant cognitive load and are faced with complex decision making responsibility in a challenging tactical environment. The overall system architecture of PerFECT framework is shown in Fig. 11.1. The overall architecture consists of four distinct elements: *Performance Evaluator*, *Feedback System*, *Curriculum Manager* and *Simulated Learning Environment (including virtual environment and custom plug-ins).*

The Performance Evaluator computes objective performance metrics by juxtaposing environmental parameters with the operator actions within the time windows framework. This data is used to identify any deficiencies based on the Skill, Rules and Knowledge Model based metrics (Rasmussen 1983) and revise the training objectives. In addition to this, the performance metrics enables real-time training scenario adaptation and is used to provide appropriate feedback to the trainees. The Feedback System allows trainee and trainer to provide feedback about the performance in any given scenario and enables feedback visualization to the trainees. To provide comprehensive feedback at overall and skill level, we targeted five basic skills namely *Movement*, *Tactical*, *Communicate*, *Technical*, and *Reporting*. The trainee's tasks were mapped into these skills to create the skills matrix that provides the link between skill level feedback and task level performance. The curriculum manager utilizes measured trainee performance at task level and scenario complexity to adapt and present additional scenarios from the training templates. The training scenario is adjusted in real-time and sent to the virtual environment. The virtual environment enables trainees to interact with the environment through a human interface (mouse and keyboard) and custom plug-in transmits environmental parameters and operator actions needed to evaluate performance back to the Performance Evaluator.

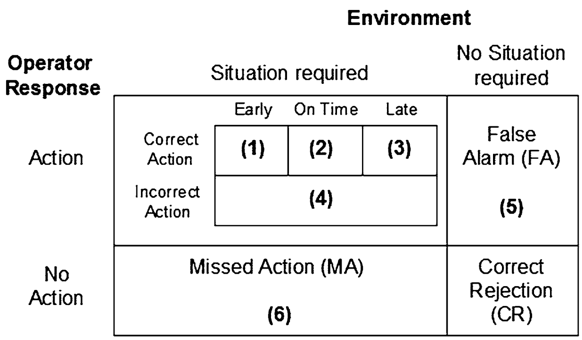


Fig. 11.2 Time window outcomes (Rothrock 2001)

11.3.2 Performance Evaluator

We used a performance evaluation framework based on Time Windows (Rothrock 2001; Thiruvengada and Rothrock 2007). Time windows are based on the fundamental theory of signal detection (Green and Swets 1988; Swets 1996) and do not prescribe a correct action that needs to be taken but rather indicate whether a trainee's action would lead to the required situation based on the current environmental demands or conditions. A time window is said to be open when an opportunity to execute an action that relates to that time window exists, or closed otherwise. A time window outcome may be classified into one of the six categories as shown in Fig. 11.2.

Time window outcomes are typically classified into six categories: *on time correct action* [area marked (2)] is defined as an action executed by the trainee that is relevant to the time window and results in the required situation; *early correct action* [area marked (1)] is defined as a trainee action that is relevant to the time window and results in the required situation, but is executed before the time window is opened; *late correct action* [area marked (3)] is defined as a trainee action that is relevant to a time window and leads to the required situation, but is executed after a time window is closed; *incorrect action* [area marked (4)] is defined as a trainee action that is relevant to a time window but does not result in the required situation; *false alarm action* [area marked (5)] is a trainee action that has no relevant time window; *missed action* [area marked (6)] is a trainee action that was not executed but the time window for that action exists.

In summary, the Performance Evaluator computes objective performance metrics by juxtaposing environmental parameters with the trainee actions within the time windows framework. More specifically, the evaluator automatically associates the trainee's actions with specific time windows that present an opportunity to perform a task. It computes performance scores by comparing the latency and accuracy of the trainee's actions with the thresholds established for "Trained", "Needs Practice" and "Untrained" subjects. The Performance Evaluator also has a user interface that displays the status of various time windows associated with specific tasks as shown in Fig. 11.3. Both measured and computed

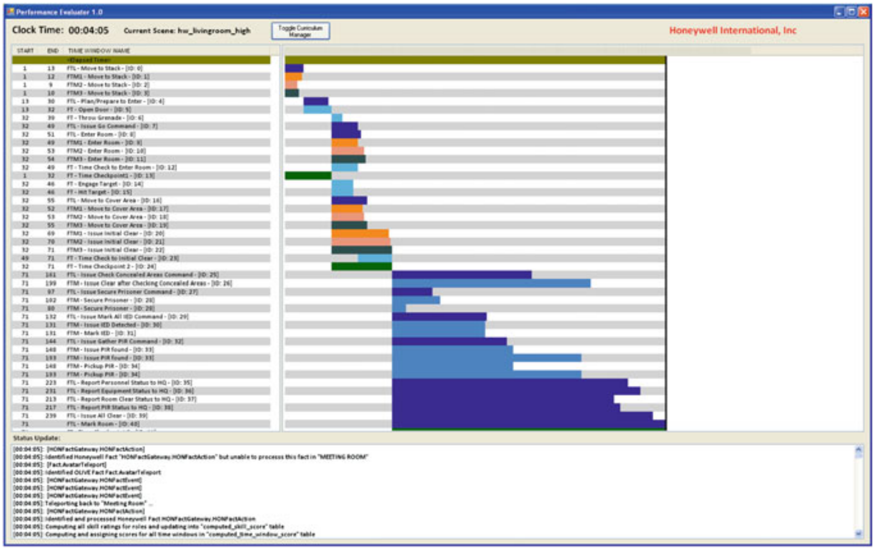


Fig. 11.3 Performance evaluator

metrics are stored in a relational database in addition to the pre-initialized optimal design data such as expert's performance thresholds.

11.3.3 Feedback System

The feedback system consists of the Feedback Server and two versions of Feedback Clients (trainer and trainee versions). Feedback server communicates with the two Feedback Clients, Performance Evaluator and relational database. Initially, the Feedback Server reads the design tables in the database to initialize the contents of the trainer's feedback Client. As the trainer observes the trainee, the trainer's Feedback Client allows the trainer to checks off all of the tasks that the trainee accomplishes as he or she completes them. Out of all tasks, only the tasks specific to the communication skill was manually evaluated by the trainer in real-time during scene execution. A time stamp is recorded for each time window on the list. After the completion of the scenario, the automated feedback system and trainee rate the performance on current scenario for each of the five skills as being "Trained," "Needs Practice," or "Untrained". In our framework, a supervising trainer can also provide feedback at skill level. In addition, the trainer indicate the completion of communication specific tasks during the scenario to compensate for the lack of an automated speech recognition system. The trainee completes the NASA-TLX (Hart and Staveland 1988) ratings, a commonly used measure of workload. Trainees receive overall as well as skill based feedback on their performance within respect to five skills as shown in Fig. 11.4.

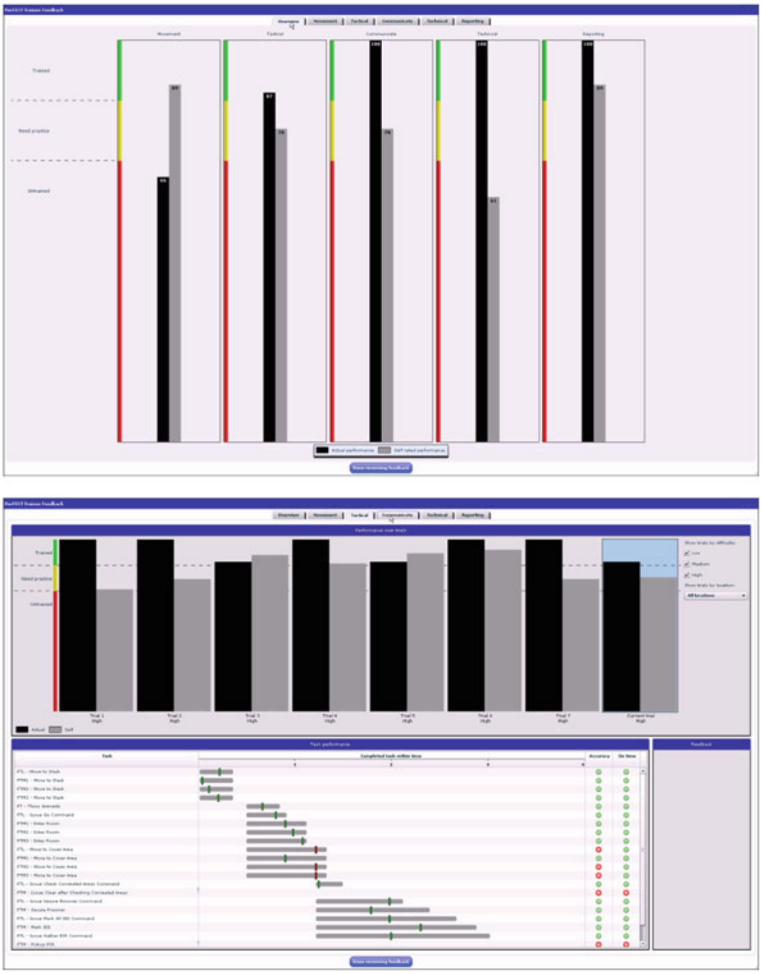


Fig. 11.4 Overall (*above*) and skill based (*below*) performance feedback

In the skill-based performance feedback, detailed information about time windows is also available. The horizontal (gray) bar at bottom of the skill based performance feedback indicates the total duration that a time window was open. A vertical performance bar that overlaps the time window bar indicates that the task was performed on time, whereas a performance bar that does not overlap the time window bar indicates that the task was not performed on time. The color coding of the bar, namely red (incorrect) or green (correct) indicates the accuracy of the action. Therefore, the trainee could easily comprehend and develop an accurate meta-cognitive understanding of how they performed in the current scenario and compare that with performance on past scenarios. For additional description on meta-cognitive feedback, readers are redirected to Tharanathan et al. (under review).

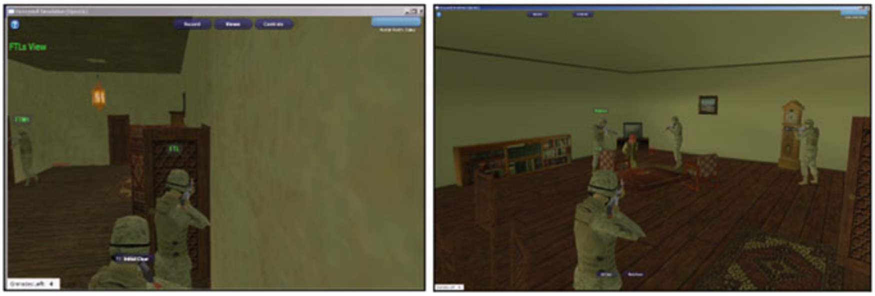


Fig. 11.5 Customized OLIVE user interface for trainer (*left*) and trainee (FTL and FTMs) (*right*)

11.3.4 Curriculum Manager

An adaptive training framework such as PerFECT requires a behavioral and cognitive model that captures the competencies of trainee's performance. PerFECT attempted to map a behavioral/cognitive model of the trainee to well established tasks and performance standards within the battle drills for the infantry rifle platoon and squad (Department of the Army 2002; Sterling and Burns 2004). These drills include task descriptions, conditions that trigger execution of task, standards of performance, as well as task steps and performance measures. The curriculum manager utilizes evaluated performance at task level and training templates (based on information received from incoming field reports that contain complete information about the actual real-world scenario) to adapt the training scenario. The training scenario is adjusted in real-time and initialized within the simulated learning environment. The curriculum consisted of scenarios that varied in complexity (low, medium and high). More specifically, the scenarios depicted a living room, a kitchen, two types of hallways and two types of bedrooms at a low, medium or high level of complexity. The level of complexity was varied by the altering the quantity of furniture, hidden or concealed areas, priority information requirements (PIR), improvised explosive devices (IED) and the number of enemy targets in the room.

11.3.5 Simulated Learning Environment: Virtual Environment and Custom Plug-ins

PerFECT framework was built to interact with the OLIVE (developed by Foterra Systems, now part of SAIC) using modular and customized plug-ins. However, the custom plug-ins can be easily modified to interface with other simulated learning environments. In addition, the user interface within OLIVE was also customized to meet the needs of the trainee's (detailed interaction capability in third person view only) and trainer (monitoring only) as shown in Fig. 11.5. Trainees control the actions and navigation of their lifelike avatar throughout the realistic 3D virtual environment with the use of a keyboard and mouse.

We also developed two specific custom plug-in components known as the Fact Digester and Fact Processor to listen to and extract relevant pieces of information relating to the trainee's action and the environment. The Fact Digester is a custom plug-in that integrates with the OLIVE server and acts as a funnel that pushes all the fact data to the Fact Processor when an action is taken by the trainee within OLIVE. On the other hand, the Fact Processor is another custom plug-in that filters and passes only the relevant facts of interest to the Performance Evaluator.

11.4 Preliminary Results

A system test was conducted using two Fire Teams (one FTL and three FTMs) and the feasibility of using the automated training framework to train the subjects was successfully demonstrated for the targeted skills using the PerFECT framework. Preliminary data analysis for the two subjects who participated in the system test is shown in Fig. 11.6. Data analysis indicates that the trainee's performance improved over time. Trainees were able to accomplish their task responsibilities with much ease even during high complexity scenarios after training with the PerFECT framework. A larger sample size is required to determine the statistical significance of these findings.

In addition to this, we also administered a survey to the two FTLs to assess the effectiveness of the PerFECT framework in informing them about their performance scores and feedback to improve the same. The results of the survey are shown in Fig. 11.7.

11.5 Future Direction

The overwhelming need for efficient and cost-effective automated training systems has motivated several efforts. However, the challenge is to develop training systems that overcome key limitations such as manual intervention, lack of adaptation, difficulty in performance assessment, use of poor performance assessment and feedback methods and slow deployment of training scenarios. We generated an automated training system that would address these needs in a positive manner. We also discovered several key insights as part of this development.

Training Design:

1. Identifying training scenarios specific to learning objectives and skills helps narrow down the mission fragments that can be effectively presented during training.

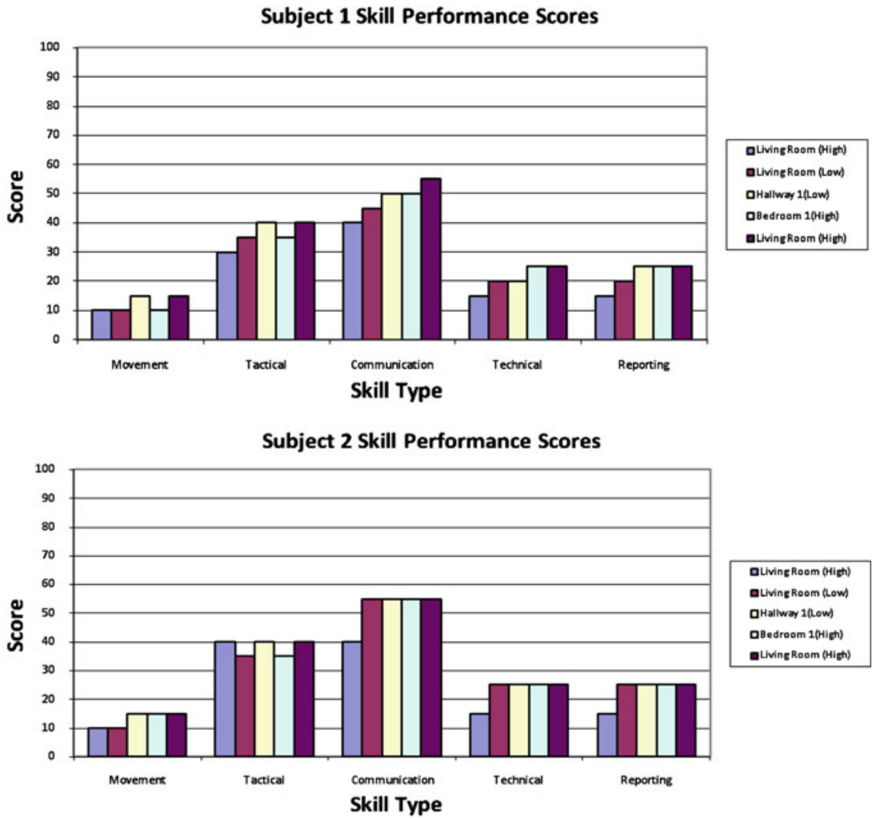


Fig. 11.6 Performance scores for each skills across all scenarios for subject 1 (*top*) and 2 (*bottom*)

2. Selecting factors that impact task performance is critical to impact trainee's learning experience and skill retention.
3. Generating scenarios from training templates with varying levels of task complexity is helpful in maintaining, managing and presenting training curriculum on the fly.
4. Creating a formalized process to map tasks to skills is very useful for creating an effective performance evaluation algorithm.
5. Formal data model is required to capture and reuse training design data more robustly.
6. Avoid presenting a large number of scenes to the trainee during a training exercise to mitigate the effects of tasks related stress and/or boredom.

System Architecture and Training System Components:

1. Using master slave client–server architecture improves control over network connection and improves data synchronization.

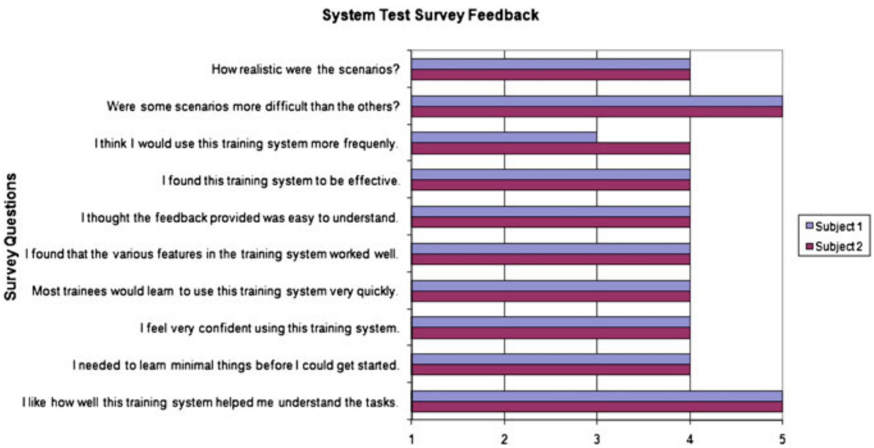


Fig. 11.7 Both subjects agreed that the training system was very useful in understanding their current performance and provide valuable insights on how to improve their performance in future sessions

2. Training support applications (such as performance evaluator, feedback system and curriculum manager) must be external to the gaming/virtual environment to allow greater control, modularity and robustness.
3. Creating visualization for displaying performance evaluation results enables monitoring of trainee's performance in real-time and increases the diagnostic value of the performance metrics.
4. Enabling trainee's to self-rate themselves and compare their rating with the computed scores helps them calibrate their own performance, as it discourages overconfidence and under-confidence issues.
5. Smart curriculum management strategies can help in training time compression and provide a better learning experience.
6. Using a flexible and realistic gaming/virtual environment is critical to providing a better learning experience. It is important to carefully consider and understand the capabilities of gaming/virtual environment while designing the software system architecture.
7. Creating custom plug-ins for data communication promotes easy tracking and evaluation of trainee's performance in real time.
8. Smart data filtering is required to extract relevant facts related to performance metrics of interest and reduce network traffic.

Acknowledgments This work supported by DARPA/IPTO under contract# HR0011-09-C-0102. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA/IPTO. We would like to thank Amy Vanderbilt (DARPA-IPTO), David Montgomery (DARPA-IPTO) and Joseph Traugott (US RDECOM-STTC) for their guidance and mentoring on this project. We would also like to thank Tim Stone (Omega Training, Cubic Corporation) for his support and shaping our understanding of the training provided to the Fire Teams. We would also like to thank members for CATT Lab (University of Maryland) including Michael Pack, Walter Lucman, Michael VanDaniker and Michael Couture for helping us understand the capabilities of the OLIVE Virtual Environment. Second Life is a registered trademark of Linden Research, Inc. OLIVE is a trademark of Forterra Systems Inc (now part of SAIC). All other trademarks used herein are the property of their respective owners. The views, opinions, and/or findings contained in this article/ presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

Anderson PL, Rothbaum BO, Hodges L (2001) Virtual reality: using the virtual world to improve quality of life in the real world. Bulletin of the Menninger Clin 65:78–91

Bainbridge WS (2007) The scientific research potential of virtual worlds. Science 317:472–476

Balzer WK, Hammer LB, Sumner KE, Birchenough TR, Martens SP, Raymark PH (1994) Effects of cognitive feedback components, display format, and elaboration on performance. Organ Behav Hum Decis Process 58:369–385

Barnett J (2009) Virtual Environments and unmanned systems: human system integration issues. In: Cohn J, Schmorrow D (eds) The PSI handbook of virtual environments for training and education: developments for military and beyond. Praeger Security International, Westport

Bewley WL, Chung GKWK, Delacruz GC, Baker EL (2009) Assessment models and tools for virtual environment training. In: Schmorrow D, Cohn J, Nicholson D (eds) The PSI handbook of virtual environments for training and education: developments for military and beyond, vol 1. Praeger Security International, Westport

Billings CE (1997) Aviation automation: the search for a human-centered approach. Erlbaum, Mahwah

Cannon-Bowers JA, Salas E (1997) A framework for developing team performance measures in training. In: Brannick MT, Salas E, Prince E (eds) Team performance assessment and measurement: theory, methods, and applications. Erlbaum, Hillsdale

Cannon-Bowers JA, Salas E (2001) Reflections on shared cognition. J Orgmet Ch 22:195–202

Cannon-Bowers JA, Bowers CA, Sanchez A (2008) Using synthetic learning Environments to train teams. In: Sessa VI, London M (eds) Work group learning: understanding, improving and assessing how groups learn in organizations. Lawrence Erbaum Associates, New York

Canon-Bowers JA, Burns JJ, Salas E, Pruitt JS (1998) Advanced technology in scenario-based training. In: Cannon-Bowers JA, Salas E (eds) Making decisions under stress: implications for individual and team training. APA, Washington

Charness N, Schultetus RS (1999) Knowledge and expertise. In: Durso FT, Nickerson RS, Schvaneveldt RW et al (eds) Handbook of applied cognition. Wiley, West Sussex

Cohn J, Nicholson D, Schmorrow D (2009) Integrated systems, training evaluations, and future directions. The PSI handbook of virtual environments for training and education: developments for military and beyond. Praeger Security International, Westport

Department of the Army (2002) Battle drills for the infantry rifle platoon and squad, ARTEP 7–8

Dieterle E, Clarke J (2008) Multi-user virtual environments for teaching and learning. In: Pagani M (ed) Encyclopedia of multimedia technology and networking, 3rd edn. Idea Group, Hershey

Endsley MR (1995) Measurement of situation awareness in dynamic systems. Hum Factors 37(1):65–84

Ericsson KA, Krampe RT, Tesch-Romer C (1993) The role of deliberate practice in the acquisition of expert performance. Psychol Rev 100:363–406

Foltz P, LaVoie N, Oberbreckling R, Rosenstein M (2009) Automated performance assessment of teams in virtual environments. In: Schmorrow D, Cohn J, Nicholson D (eds) The PSI handbook of virtual environments for training and education: developments for military and beyond, vol 1. Praeger Security International, Westport

Gordon SE (1994) Systematic training program design: maximizing effectiveness and minimizing liability. Prentice Hall, Englewoods Cliffs

Green DM, Swets JA (1988) Signal detection theory and psychophysics. Peninsula Publishing, Los Altos

Grier RA, Warm JS, Dember WN, Matthews G, Galinsky TL, Szalma JL, Parasuraman R (2003) The vigilance decrement reflects limitations in effortful attention, not mindlessness. Hum Factors 45:349–359

Grubb PL, Warm JS, Dember WN, Berch DB (1995) Effects of multiple-signal discrimination on vigilance performance and perceived workload. Hum Factors Ergon Soc Annu Meet Proc 39:1360–1364

Hart SG, Staveland LE (1988) Development of NASA-TLX (Task Load Index): results of experimental and theoretical research. In: Hancock PA, Meshkati N (eds) Human Mental Workload. North Holland Press, Amsterdam pp 139–183

Johnston JH, Smith-Jentsch KA, Cannon-Bowers JA (1997) Performance measurement tools for enhancing team decision making. In: Brannick MT, Salas E, Prince C (Eds) Assessment and measurement of team performance: Theory, research, and applications. NJ: Erlbaum, Hillsdale pp 45–62

Kirwan B, Ainsworth LK (1992) A guide to task analysis. Taylor & Francis, Bristol

Kluger AN, DeNisi A (1996) The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. Psychol Bull 119:254–284

Kozlowski SWJ, Toney RJ, Mullins ME, Weissbein DA, Brown KG, Bell BS (2001) Developing adaptability: a theory for the design of integrated-embedded training systems. In: Salas E (ed) Advances in human performance and cognitive engineering research, vol 1. JAI/Elsevier Science, Amsterdam

Lampton DR, Bliss JP, Morris CS (2002) Human performance measurement in virtual environments. In: Stanney KM (ed) Handbook of virtual environment: design, implementation, and applications. Erlbaum, Mahwah

Lee J, Moray N (1992) Trust, control strategies and allocation of function in human-machine systems. Ergonomics 35(10):1243–1270

Lewandowsky S, Little DR, Kalish M (2007) Knowledge and expertise. In: Durso FT, Nickerson R, Dumais S, Lewandowsky S, Perfect T (eds) Handbook of applied cognition, 2nd edn. Wiley, West Sussex

Loomis JM, Blascovich JJ, Beall AC (1999) Immersive virtual enviornment technology as a basic research tool in psychology. Behav Res Methods instrum comput 31:557–564

Mantovani F (2001) VR learning: potential and challenges for the use of 3D environments in education and training. In: Riva G, Galimberti C (eds) Towards cyberpsychology. IOS Press, Amsterdam

Meister D (2004) Conceptual foundations of human factors measurement. Earlbaum, Mahwah

Moray N, Inagaki T, Itoh M (2000) Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. J Exp Psychol Appl 6:44–58

Mosier KL, Skitka LJ, Heers S, Burdick M (1997) Automation bias: decision making and performance in high-tech cockpits. Int J Aviat Psychol 8:47–63

Nicholson D, Schmorrow D, Cohn J (2009) The PSI handbook of virtual environments for training and education: developments for military and beyond, vol 2. VE components and training technologies. Praeger Security International, Westport

Oser RL, Gualtieri JW, Cannon-Bowers JA, Salas E (1999) Training team problem-solving skills: an event-based approach. Comp Hum Behav 15:441–462

Parasuraman R, Riley V (1997) Humans and automation: use, misuse, disuse, abuse. Hum Factors 39:230–253

Rasmussen JR (1983) Skills, rules and knowledge: signals, signs, symbols, and other distinctions in human performance models. IEEE trans syst man cybern 13:257–266

Rasmussen J (1986) Information processing and human-machine interaction: an approach to cognitive engineering. Elsevier, New York

Riva G (1997) Virtual reality in neuro-psycho-physiology: cognitive, clinical and methodological issues in assessment and rehabilitation. IOS Press, Amsterdam

Roman PA, Brown D (2008) Games—just how serious are they? Interservice/industry training, simulation and education conference (I/ITSEC). Academic Press, Orlando

Rothrock L (2001) Using time windows to evaluate operator performance. Int J Cogn Ergon 5:1–21

Salas E, Cannon-Bowers JA (1997) Methods, tools, and strategies for team training. In: Quinones MA, Ehrenstein A (eds) Training for a rapidly changing workplace: applications of psychological research. APA, Washington

Salas E, Oser RL, Cannon-Bowers JA, Daskarolis-Kring E (2002) Team training in virtual environments: an event-based approach. In: Quiñones MA, Ehrensstein A (eds) Training for a rapidly changing workplace: applications of psychological research. APA, Washington

Salas E, Cannon-Bowers JA (2001). The science of training: A decadeof progress. Ann Rev Psychol 52:471–499

Salas E, Priest HA, Wilson KA, Burke CS (2006) Scenario-based training: improving military mission performance and adaptability. In: Adler AB, Castro CA, Britt TW (eds) Military life: the psychology of serving in peace and combat. Praeger Security International, Westport

Sarter NB, Woods DD (1997) Team play with a powerful and independent agent: operational experiences and surprises on the Airbus A-320. Hum Factors 39:553–569

Schmorrow D, Cohn J, Nicholson D (2009) The PSI handbook of virtual environments for training and education: developments for military and beyond, vol 1. Learning, requirements and metrics. Praeger Security International, Westport

Schultetus S, Charness N (1999) Recall vs position evaluation revisited: the importance of position-specific memory in chess skill. Am J Psychol 112(4):555–569

Shaffer DW, Resnick M (1999) “Thick” authenticity: new media and authentic learning. J Interact Learn Res 10:195–215

Smithers JW, Wohers AJ, London M (1995) A field study of reactions to normative versus individualized upward feedback. Group Organ Manag 20:61–89

Sterling BA, Burns CA (2004) Skills required for platoon leaders in the objective force unit of action. Army Research Lab Aberdeen Proving Ground, MD

Swets JA (1996) Signal detection theory and ROC analysis in psychology and diagnostics: collected papers. Earlbaum, Mahwah

Tharanathan A, Derby P, Thiruvengada H (under review) Training for metacognition in simulated environments. In:Human-in-the-Loop Simulations. Springer, London

Thiruvengada H, Rothrock L (2007) Time windows based team performance measures: a framework to measure team performance in dynamic environments. Cogn Tech Work 9:99–108

van Buskirk W, Cornejo J, Astwood R, Russell S, Dorsey D, Dalton J (2009) A theoretical framework for developing systematic instructional guidance for virtual environment training. In: Schmorrow D, Cohn J, Nicholson D (eds) The handbook of virtual environment training. Praeger Security International, Westport

van Dongen KW, Tournoij E, van der Zee DC, Schijven MP, Broeders IAMJ (2007) Construct validity of the LapSim: can the LapSim virtual reality simulator distinguish between novies and experts? Surg Endosc 21:1413–1417

Verdaasdonk EG, Stassen LP, Monteny LJ, Dankelman J (2005) Validation of a new and simple virtual reality simulator for training of basic endoscopic skills. Surg Endosc 20:1–9

Chapter 12
Evaluating Human Interaction with Automation in a Complex UCAV Control Station Simulation Using Multiple Performance Metrics

Sasanka Prabhala, Jennie J. Gallimore and Jesse R. Lucas

Abstract The dynamics and complexities of human–machine systems and the overwhelming amount of data that must be handled by human operators is making automation a critical factor in planning, decision-making, and in execution in many complex systems. Complex systems are characterized by uncertainty, ambiguity, ill-defined goals, dynamically changing conditions, distractions, and time pressures. A semi-autonomous system requiring significant human-centered design support are remotely operated vehicles (ROVs) such as unmanned aerial vehicles (UAVs), unmanned combat aerial vehicles (UCAVs), space maneuverable vehicles (SMVs), and unmanned emergency vehicles (UEVs). The objectives of this research are to (1) develop a simulation system that would allow investigation of human operator performance issues when supervising multiple UCAV vehicles, and (2) investigate human performance through the collection of multiple dependent measures. The simulation tool was designed to be adaptable to allow continued research on a variety of factors related to the control of autonomous vehicles. A research study using this simulation tool investigated the effects of automation and the number of UCAVs being controlled on operator performance during an identify and destroy mission. Results indicate that increasing the number

S. Prabhala (✉)
Interactions and Experience Research Group, Intel Labs,
2111 NE 25th Avenue, Hillsboro, OR 97124, USA
e-mail: sasanka.v.prabhala@intel.com

J. J. Gallimore · J. R. Lucas
Department of Biomedical, Industrial, and Human Factors Engineering,
Wright State University, 207 Russ Engineering Center,
3640 Col. Glenn Hwy, Dayton, OH 43435, USA
e-mail: jennie.gallimore@wright.edu

J. R. Lucas
e-mail: jesse.r.lucas@wright.edu

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6_12,

of UCAVs significantly increased workload under LOW Automation and the increase in workload was reduced when HIGH Automation was introduced. This study showed that operators are able to control multiple UCAVs more effectively with appropriate automation.

12.1 Introduction

The dynamics and the complexity of semi-autonomous systems such as command and control of ROV and the overwhelming amount of data that must be handled by the human operators makes automation a critical factor in planning, decision-making, and execution. With the introduction of automated aids, the role of the human operator in human-centered automation has changed from an active controller to that of a passive controller responsible to ensure a safe outcome following unforeseen circumstances. This change in the role of the controller has adverse effects on the performance of the system such as vigilance decrements (Nathalie et al. 2008; Warm et al. 2008), out-of-the-loop performance problems (Barnes and Matz 1998; Endsley and Garland 1999), trust biases (Alberdi et al. 2009), complacency (Jessie and Barnesl 2008; Parasuraman and Hancock 2008), reliability (Dixon and Wickens 2004), skill degradation (Mooij and Corker 2002; Cummings and Mitchell 2008), and attention biases (Sarter and Woods 1995; Mosier and Skitka 1996).

Warm et al. (2008) demonstrated that when operators are passive monitors, the ability of the human controller to process information and detect the need for intervention is slowed. Even if the operator is able to function as an effective monitor, the use of automation can lead to significant problems in taking over the system control, during automation failure (Ruff et al. 2002). Over reliance on automated aids not only affect immediate awareness (Endsley 1996; Taylor 2002), but can also lead to long-term skill degradation (Parasuraman et al. 2000). In addition, human operators working with semi-autonomous systems is much like working with a partner having limited interaction capabilities. For instance, you may only ask the partner pre-selected questions. The partner can only express its thoughts via pre-selected media and formats. Rephrasing an answer or statement of information is usually not an option. Data must be translated into information that is relevant to the operator's current needs. The operator must also remember how and why the automation acts or reacts to situations based on training and system use.

Many of these problems between the human operator and the automated tools in the command and control of ROVs arise due to incomplete or inappropriate understanding of automation cues. This can be attributed to the fact that automated systems do not provide meaningful information which can be used by the human operator to make decisions that lead to enhanced system performance. Improving the operational effectiveness of semi-autonomous systems is a critical area of research, as humans are the end users of the system and perform important tasks and make critical decisions to fulfill mission requirements.

Although the human is removed from direct control in automated systems such as UAVS, UCAVs, SMVs, and UEVs, the human operators from remote locations perform supervisory control tasks such as monitoring, planning, control, and coordination. Most of the current ROV designs are either too "vehicle centered" or rely heavily on commercial of the shelf (COTS) technology to reduce costs and time. Often technological advances in hardware occur so rapidly that the technology centered approach pushes system designers to design systems without evaluating variables or system parameters that can affect human performance.

Initial ROV reconnaissance missions were controlled by two or more operators (Dixon et al. 2004). However, systems are now being designed such that a single operator will be able to control many (Cummings et al. 2007). If a single operator is to handle multiple UCAVs in a combat setting, many features will have to be automated to assist the operator. Hence the critical issues in the design of these systems are (1) determining the appropriate levels of autonomy, (2) interaction of the human operator with the intelligent UCAVs and (3) the interface of the human controller system.

Automation can be implemented at various levels (Sheridan 1980; Endsley 1987 and Wickens 1999) and does not exist in all or none fashion (Meyer et al. 2003). Particularly deficient are systematic approaches to allocating functions between humans and automation, studies of coordination between remote operators and intelligent controller nodes in the unmanned vehicles, and studies of human/system interface development in remote command and control. The outcome is the development of systems that are very difficult for human operators to supervise. As the systems engineering process unfolds for these promising new systems, the combination of air vehicle technologies, weapon capabilities, and operational tactics must be integrated with the control interface based on a comprehensive understanding of the human's capabilities and limitations during tradeoffs conducted prior to a design freeze. Hence, there is a need to study human performance in complex systems such as ROV domains to make an impact on the design process.

Therefore, it is important to develop a simulation system that will facilitate effective evaluation of the effects of system parameters and automation on human decision-making in ROVs. We designed and developed a simulation system that allows systematic analyses of human operator performance issues when supervising multiple remotely operated vehicles and also presents the information in a cognitively effective manner to keep the human in the loop. The remainder of the chapter details the implementation of the simulation architecture, and discusses the results of empirical evaluation to investigate human performance through the collection of multiple dependent measures as the automation level and number of vehicles supervised changed. The domain we investigated is the command and control of UCAVs in suppression of enemy air defence (SEAD) mission.

Table 12.1 Parameters monitored and controlled by human operators

1. Number of UCAVs
2. Selection and deletion of waypoints along mission route
3. UCAV parameters of airspeed, altitude, and fuel consumption
4. Selection of sensors for identification of enemy or friendly targets
5. Level of automation of tasks including:
• Identification of targets
• Adding new waypoints
• Monitoring UCAV to target ranges
• Selecting proper munitions
• Selecting proper sensors
• Bringing priority targets to focus
6. Remove or display the UCAV routes or targets (ability to declutches)
7. Reduce or increase background contrast

12.2 Interface and Simulation

A UCAV control interface was developed to run a simulated SEAD mission written in Java® for the Windows platform. This simulation interface allows operators to monitor and control multiple UCAVs travelling along various waypoints in order to destroy enemy defences. Table 12.1 lists the parameters that can currently be monitored or controlled by operators using this software.

The simulation architecture is modular and allows easy manipulation of system variables such as relative speed, map backgrounds, fuel consumption rate, symbology, number of UCAVs, and level of automation to support rapid prototyping for future studies. Also, the simulation was designed to collect multiple performance parameters to help determine the sensitivity of various dependent measures. The performance parameters collected include correct target identifications, targets not destroyed or identified, non-targets destroyed, UCAV return time, average time in target range, and times in target range without taking action.

The simulation involves monitoring and controlling UCAVs within an enemy territory populated by several known and unknown targets. The underlying simulation architecture supports active human interaction where the user interface displays the state of the system and also facilitates the change of the simulated system state during execution.

The interface and simulation included up to 4 UCAVs that travel along individual predetermined flight paths, as illustrated in Fig. 12.1. The flight paths are made of waypoints connected by lines. Waypoints are destinations on the map display. Each UCAV moves from one waypoint to the next along the lines connecting the waypoints. Waypoints can be moved, added, or deleted by the operator. Waypoints can be used to set a UCAV's airspeed and altitude as well as heading. Airspeeds, altitudes, and flight paths are all pre-programmed before a mission begins, but can be changed at any time during the mission.

As the UCAVs fly along their individual routes the operator monitors and controls the UCAVs in order to identify unknown targets, destroy enemy targets,

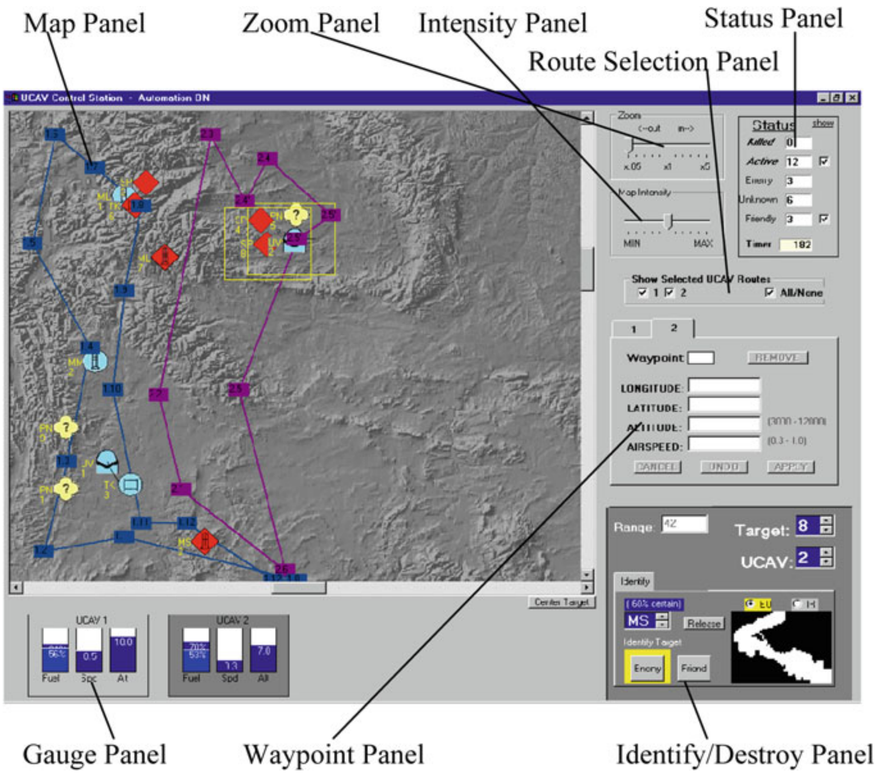


Fig. 12.1 Snapshot of simulation interface

and return the UCAVs to base safely. As the UCAVs fly along their route, targets are detected via satellites, friendly ground radar, sensors on the UCAVs, or other sources. The targets randomly appear at random locations on the map as the UCAV control station database is updated by all these sources.

Onboard each UCAV there are two sensors used for viewing targets remotely. One is an infrared (IR) sensor and the other is an electro optical (EO) sensor. Each can be used for looking at different types of unknown targets. UCAVs must be within the range of 5000 ft (about 1 mile) from an unknown target to view the target with either sensor and if needed destroy the target with specified ammunition. Range is measured in hundreds of feet; thus 5000 ft is displayed as a range of 50.

Each UCAV has an unlimited supply of 4 different types of ammunitions. Different ammunitions are used for destroying different types of targets. Any type of target, other than friendly, may put a radar lock on a UCAV any time it is in range. This means that the target can track the exact UCAV location and can cause damage to the UCAV. Since it was a controlled experiment and destroying or damaging UCAVs would change the number of UCAVs controlled by human operators and would confound our empirical analyses, a penalty function was assigned when each time a target puts a radar lock on the UCAV without actually destroying the UCAV. At any time a UCAV could lose altitude or airspeed and fall below the proper range. It is the operator's task to monitor and adjust each UCAV's airspeed and altitude to be sure that it stays within the proper range.

Since the operator uses the interface to interact with the problem, the automation must be able to effectively communicate information about the state of the system to the operator. The different ways in which information is presented to the operator through automation aids, need to be tailored to the type of environment and task at hand. According to Moiser's and Skitka (1996) the broad range of the capabilities of automated decision aids include displaying or highlighting raw data that needs to be attended to, providing system monitoring and alert signals, displaying trends and the results of diagnosis with confidence information, displaying potential predicted consequences for a particular course of action, providing wrong decision protection, and providing action directive.

Our interface was designed to support Mosier's and Skitka's (1996) six levels of information presentation to involve human-in-the-loop decision making for improved human performance. The association between our interface and Mosier's and Skitka's (1996) capabilities of automation are listed below:

- *Display or highlight raw data that needs to be attended to*: The remaining mission time is shown continuously through the SEAD mission. Controllers translate this data into the amount of time to move from one waypoint to another and the amount of time to spend on each target (depending on its priority). Also, targets are automatically queued when they are within range of a UCAV.
- *Provide system monitoring and alert signals*: UCAVs and targets are outlined in red when a target has a UCAV in radar lock. Altitude and airspeed are monitored and highlighted in red when out of safe flying boundaries.
- *Display trends and the results of diagnosis with confidence information*: Automatic identification of targets is accompanied by an accuracy percentage (i.e. probability of being correct).
- *Display potential predicted consequences for a particular course or action*: The fuel gauge not only displays the amount of fuel left at a particular instant, but also displays the estimated amount of fuel left at the estimated time of that UCAV's return to the base. This allows the operator to see the consequences of extending the flight path or increasing speed along a segment of the path.
- *Provide wrong decision protection*: As new targets appear, waypoints are automatically added to the flight paths, but operators are still able to adjust those waypoints to better approach the target or fit mission limitations (e.g. time).
- *Provide action directive*: When targets are automatically queued unknown targets are queued such that the identification panel appears, and already identified targets are cued such that the ammunition selection (i.e. destroy panel) panel appears, thus giving direction to the type of action that needs to be accomplished for the particular target. In addition, the proper ammunition or sensor is automatically selected (100% accurate), and targets are automatically identified (75% accurate).

Table 12.2 Automated Items under HIGH and LOW Automation

LOW automation	HIGH automation
1. UCAVs automatically follow the indicated flight paths, at the indicated airspeeds and altitudes 2. Altitude and airspeed are monitored and highlighted in red when out of safe flying boundaries	1–6 Same as LOW Automation 7. As new targets appear, waypoints are automatically added to the flight paths, but operators are still able to adjust those waypoints to better approach the target or fit mission limitations (e.g. time) 8. Targets are automatically queued when they are within range of a UCAV.
3. The fuel gauge not only displays the amount of fuel left at a particular instant, but also displays the estimated amount of fuel left when the UCAV returns to the base	9. When targets are automatically queued unknown targets are queued such that the identification panel appears, and already identified targets are cued such that the ammunition selection (i.e. destroy panel) panel appears, thus giving direction to the type of action that needs to be accomplished for the particular target
4. Sensors are automatically rotated and focused when selected for viewing specific targets 5. Remaining mission time is tracked and shown continuously through the SEAD mission	10. The proper ammunition or sensor is automatically selected (100% accurate), and targets are automatically identified (75% accurate) when targets are cued. The operator must decline, accept, or adjust both destroy and identification actions
6. UCAV and target are outlined in red when a target has a UCAV in radar lock	11. Automatic identification of targets is accompanied by an accuracy percentage (i.e. probability of being correct)

12.3 Human Performance Evaluation

12.3.1 Experimental Design

12.3.1.1 Independent Variables

The simulation was used to investigate the effects of level of automation and number of UCAVs supervised on human performance. There were two levels of automation investigated. Table 12.2 lists the capabilities of both levels of automation (LOW and HIGH). The second independent variable was the number of UCAVs controlled: 1, 2 or 4. The experimental design was a 2 $\times$ 3 within-subjects design.

12.3.1.2 Dependent Variables

For this study we evaluated multiple objective measures of performance that would provide an indication of mission success. Ten dependent measures were evaluated as defined below.

- *Portion of targets correctly Identified (Correct ID)*: The ratio of targets correctly identified divided by the total number of targets to be identified in a trial.
- *Portion of targets not Identified (No ID)*: The ratio of targets that were not identified divided by the total number of targets to be identified in a trial.
- *Portion of targets incorrectly Identified (Incorrect ID)*: The ratio of targets incorrectly identified divided by the total number of targets to be identified in a trial.
- *Portion of enemy targets destroyed (Enemy Destroyed)*: The ratio of targets destroyed divided by the total number of targets to be destroyed in a trial.
- *Portion of enemy tanks destroyed (Tanks Hit)*: The ratio of tanks destroyed divided by the total number of tanks present in a trial. Tanks were considered non-threats and participants were instructed not to waste ammunition on tanks, however, the automation could choose a tank to destroy requiring the operator to notice the type of target and decline the action.
- *Number of times in range without taking action (In Range)*: Number of times a UCAV entered into range of a target that needed to be identified or destroyed and neither action was taken.
- *Average Time of Non-Radar Range Faults*: A non-radar range incident is when a UCAV enters into range of a target that does not put the UCAV into radar lock. The average time (sec) of non-radar range incidents is the total time UCAVs were in non-radar range divided by the number of times they were in non-radar range.
- *Average Time of an Altitude Fault*: The average length of time in seconds a UCAV was below the recommended altitude.
- *Portion UCAVs Returned on Time*: The ratio of UCAVs returned to base within the targeted return time divided by the total number of UCAVs that need to be returned to the base.
- *Simulation Score*: The simulation score is a weighted average of the participant's performance for a trial based on the prioritized rule list summarized in Table 12.3. To accurately compare differences between controlling 1, 2, and 4 UCAVs the total score is divided by the number of UCAVs in that trial to obtain the individual UCAV score. A large UCAV score indicates better performance.

12.3.2 Subjects

Sixteen students from Wright State University were paid for their participation. All participants were screened for normal or corrected 20/20 vision and color vision capabilities. This criterion was important as participants should be able to differentiate between types of targets, track the UCAVs' altitude and airspeed, and monitor different UCAVs.

Table 12.3 Rule List for obtaining Score during Trials

1.	Identification of specific targets	100 points for correctly identified, −120 for targets identified incorrectly, −100 for targets not identified at the end of the mission
2.	Destruction of specific targets	80 points for destroyed target, −80 points for destroyed tanks, −80 for enemy targets not hit at the end of the mission
3.	Do not fly into target range longer than is necessary	−10 points for every second in target range
4.	Return to base with fuel in each UCAV tank	−200 points when run out of fuel and −5 points for every second for out of safe flying boundary
5.	Maintain proper altitude for each UCAV	−5 points for every second not at proper altitude
6.	Maintain proper airspeed for each UCAV	−5 points for every second not at proper airspeed
7.	Do not allow any UCAV to return to base early (45 s mark), but return before time is out	−200 points if returned to base early or late
8.	Do not allow the target to put a lock on the UCAV	−100 points each time the target has a radar lock on the UCAV
9.	Use proper munitions against different types of targets	Will not destroy if wrong munitions are used
10.	Use proper sensor to identify different types of unknown targets	Will not show image without proper sensor

12.3.3 Apparatus

The simulation was written in Java® and run on a personal computer with Windows OS. A 17-inch monitor was used to display the interface, and a mouse and keyboard were used as input devices. The experiment took place in an office type environment with dim lighting. The room was a sound proof room that allowed for no distractions. The participants sat in an adjustable office chair, and the keyboard and mouse were placed at comfortable positions determined by each participant.

12.3.4 Procedure

The participants were trained on the objectives of the SEAD mission and the use of the control interface. Tasks taught during training include detecting, identifying, and destroying targets; manipulating the altitude, direction, and speed of the individual UCAVs; and monitoring time and mission progress. Training lasted one hour. Participants were given a 5-min break after training.

Upon completion of training the participant ran through six practice mission simulations that were set up exactly like the actual experimental simulations.

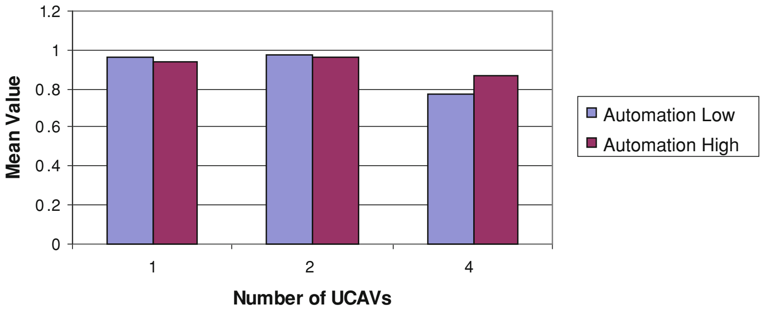


Fig. 12.2 Portion of targets correctly identified as a function of number of UCAVs and automation level

Each mission lasted 8 min. Time was displayed for the participants in seconds from '480' counting down to '0'. Each UCAV began the mission at its respective waypoint '1.1' at the bottom center of the map display area. Participants were presented with 1, 2, or 4 UCAVs and initial paths indicated by waypoints. Random targets popped up in random locations at random times.

The participant could delete waypoints, add waypoints, and, for any unvisited waypoints, change a waypoint's position, change the altitude of the UCAV to the waypoint, and change the airspeed of the UCAV to the waypoint. The participant participated in six 8 min trials. During any given trial the participant monitored the assigned number of UCAVs, identified specified targets, and responded appropriately to specified targets (identify, destroy, ignore). Responses include mouse clicks, mouse drags, and keyboard data entry.

12.3.5 Results

Participants' performance was separately analyzed via planned analysis of variance (ANOVA) for each separate dependent variable. The alpha criterion was set to 0.05. Post-hoc Tukey's tests were run to analyze significant interactions and main effects. The Statistical Analysis System (SAS) software was used for analysis.

12.3.5.1 Portion of Targets Correctly Identified (Correct ID)

Figure 12.2 illustrates the significant interaction between Automation and Number of UCAVs $F(2, 28) = 5.56$, $p = 0.0093$. Posthoc Tukey's tests indicated that when participants were controlling 4 UCAVs under LOW Automation the portion of targets correctly identified ($X = 0.775$) was significantly lower than all other conditions except when controlling 4 UCAVs with HIGH Automation

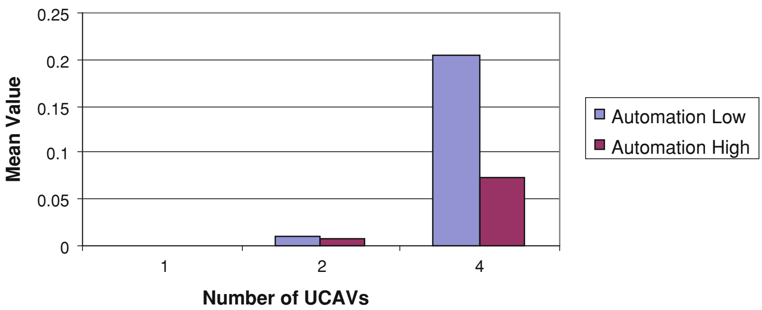


Fig. 12.3 Portion of targets not identified as a function of number of UCAVs and automation level

(X = 0.871). When controlling 4 UCAVs under HIGH Automation performance is not significantly different from controlling 1 or 2 UCAVs under either level of automation. There are no other significant differences.

12.3.5.2 Portion of Targets Not Identified (No ID)

The results indicated a significant main effect for Automation $F(1, 14) = 9.83$, ($p = 0.0073$). A significantly greater portion of targets were left unidentified with LOW Automation (X = 0.072) than with HIGH Automation (X = 0.028). The results indicated a significant main effect for Number of UCAVs $F(2, 28) = 32.79$, ($p \leq 0.0001$). A significantly greater portion of targets were left unidentified when 4 UCAVs were controlled (X = 0.14) than when 1 or 2 UCAVs were controlled (X = 0 and 0.09 respectively).

The main effects are modified by a significant interaction between Automation and Number of UCAVs $F(2, 28) = 8.67$, ($p = 0.0012$) illustrated in Fig. 12.3. When subjects were controlling 4 UCAVs under LOW Automation the portion of targets not identified (X = 0.204) was significantly higher than all other conditions. Without the HIGH Automation, the workload for 4 UCAVs affects the operator's ability to identify a higher portion of unknown targets. When subjects were controlling 4 UCAVs under HIGH Automation the portion of targets not identified (X = 0.076) was significantly different from when subjects were controlling 1 UCAV under either HIGH or LOW Automation (X = 0 for both).

12.3.5.3 Portion of Targets Incorrectly Identified (Incorrect ID)

The results indicated a significant main effect for Automation $F(1, 14) = 5.37$, ($p = 0.0361$). A significantly greater portion of targets were identified incorrectly

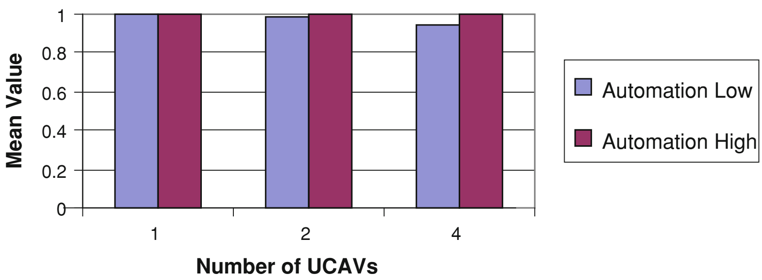


Fig. 12.4 Portion of enemy targets destroyed as a function of number of UCAVs and automation level

with HIGH Automation (X = 0.068) compared to LOW Automation (X = 0.021). There were no other significant effects.

12.3.5.4 Portion of Enemy Targets Destroyed (Enemy Destroyed)

The results indicated a significant main effect of Automation $F(1, 14) = 8.19$, ($p = 0.0126$). Posthoc Tukey's tests indicated that a significantly greater portion of enemy targets were destroyed with HIGH Automation (X = 0.999) than with LOW Automation (X = 0.98). The results indicated a significant main effect for Number of UCAVs $F(2, 28) = 4.05$, ($p = 0.0284$). A significantly greater number of targets were destroyed when 1 UCAV is controlled (X = 100) than when 4 UCAVs are controlled (X = 0.973).

Figure 12.4 illustrates the significant interaction between Automation and Number of UCAVs $F(2, 8) = 3.70$, $p = 0.0377$. There was a significantly lower portion of enemy targets destroyed when controlling 4 UCAVs with LOW Automation (X = 0.951) compared to all other conditions. Without HIGH Automation, the workload for 4 UCAVs affects the operator's ability to destroy a higher portion of enemy targets. In general, performance was very high for this dependent variable indicating little or no difficulty with this task.

12.3.5.5 Portion of Tanks Destroyed (Tanks Hit)

A main effect for Automation was found to be significant $F(1, 14) = 11.89$, ($p = 0.0039$). A significantly greater portion of tanks were destroyed when the Automation level was HIGH. A main effect for Number of UCAVs was also significant $F(2, 28) = 3.85$, ($p = 0.0335$). A significantly greater proportion of tanks were destroyed when 4 UCAVs were controlled than when 1 UCAV was controlled.

The main effects were modified by the significant interaction between Automation and Number of UCAVs, $F(2, 28) = 3.85$, ($p = 0.0335$). Under LOW

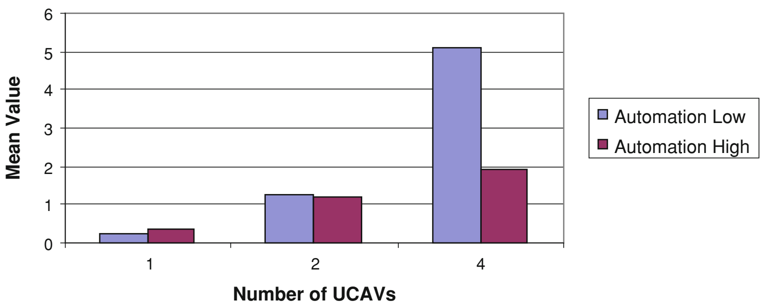


Fig. 12.5 Number of times UCAV are in range without action as a function of number of UCAVs and automation level

Automation conditions, and under HIGH Automation with 1 UCAV, no enemy tanks were destroyed. When automation was HIGH and 2 or 4 UCAVs were being controlled, enemy tanks were hit (X = 0.06 and X = 0.20 respectively). Posthoc Tukey's tests indicated that the condition of 4 UCAVs under HIGH Automation (X = 0.20) resulted in significantly higher portion of tanks destroyed. Recall that the rule was to avoid targeting tanks.

12.3.5.6 Number of Times Each UCAV Was in Range of a Target Without Taking Action (In Range)

A main effect for Number of UCAVs was found to be significant $F(1, 14) = 4.94$, ($p = 0.0146$). The average number of times a UCAV entered into range of a target without taking action was significantly greater when controlling 4 UCAVs (X = 0.883) than when controlling 1 UCAV (X = 0.313).

The total number of times during a trial that all the UCAVs entered into range of targets that needed to be destroyed or identified and neither action was taken was divided by the number of UCAVs controlled during that trial to normalize the data. Figure 12.5 illustrates the significant interaction between Automation and Number of UCAVs $F(2, 28) = 6.17$, ($p = 0.0060$). Posthoc tests indicate that when subjects controlled 4 UCAVs with LOW Automation the average number of times a UCAV was in range without taking action (X = 1.281) was significantly higher than all other conditions except when controlling 2 UCAVs with LOW Automation (X = 0.625). Controlling 2 UCAVs with LOW Automation was not significantly different fromany other condition.

12.3.5.7 Average Time of Non-Radar Range Faults

A non-radar range incident is when a UCAV enters into range of a target that does not put the UCAV into radar lock. The average time (sec) of non-radar range

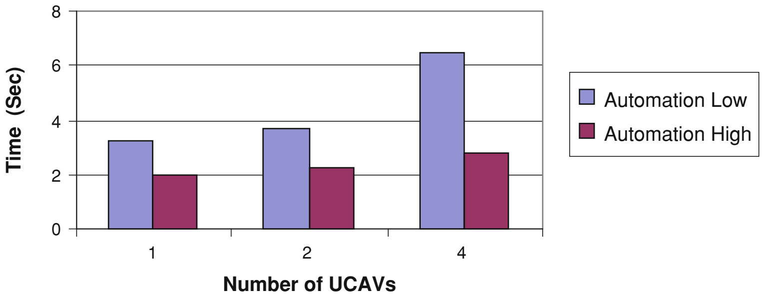


Fig. 12.6 Average time (sec) of non-radar range faults, a small time indicates better performance

incidents is the total time UCAVs were in non-radar range divided by the number of times they were in non-radar range. Fig. 12.6. illustrates the significant interaction between Automation and Number of UCAVs $F(2, 28) = 10.17$, ($p = 0.0005$). Under LOW Automation, the average time of non-radar range faults increased as the number of UCAVs increased. Posthoc Tukey's tests indicated that there was a significant difference between controlling 4 UCAVs ($X = 6.488$) compared to controlling 1 ($X = 3.242$) or 2 UCAVs ($X = 3.710$) under LOW Automation. Under the HIGH Automation condition, performance was similar regardless of the Number of UCAVs being controlled.

A significant main effect for Automation was found $F(1, 14) = 60.86$, ($p \leq 0.0001$). With LOW Automation the average time of a non-radar range incident was significantly higher ($X = 1.963$) than with HIGH Automation ($X = 1.154$). A significant main effect for Number of UCAVs was also found $F(2, 28) = 34.42$, ($p \leq 0.0001$). With 4 UCAVs to control, the average time of non-radar range incident was greater ($X = 4.631$) than when 1 ($X = 2.592$) or 2 ($X = 2.979$) UCAVs were controlled.

12.3.5.8 Average Time of an Altitude Fault

The average time of an altitude fault is the average length of time (sec) during a trial that a UCAV was below the recommended altitude. There was a significant main effect of Number of UCAVs for the average time of an altitude fault. With 4 UCAVs to control the average time of an altitude fault ($X = 16.723$) was greater than when controlling 1 UCAV ($X = 1.844$). These results indicate that as the Number of UCAVs increase the operators paid less attention to the altitude levels of the UCAVs, and thus took much longer to respond to altitude faults.

Although the number of altitude faults depends on how many UCAVs are being controlled an analysis of average time is useful in comparing within a given mission area. Assume a given mission area requires 4 UCAVs to patrol the area. Either 4 individual operators can monitor 1 UCAV each or 1 operator can monitor

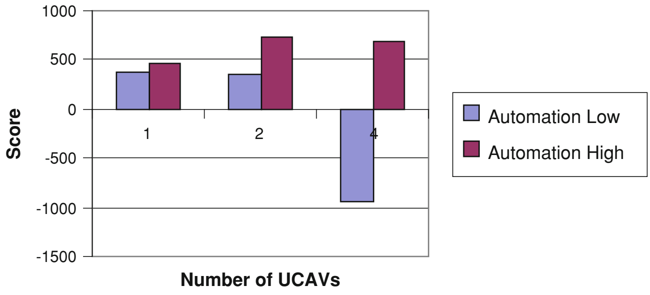


Fig. 12.7 The Simulation score is a weighted average of the participant's performance for a trial based on the prioritized rule list (summarized in Table 12.3); a large score indicates better performance

4 UCAVs within the same mission area. Our results indicate that when one operator controls 4 UCAVs the average altitude fault is 16.723 s. However, if there were 4 operators controlling 1 UCAV each covering the same area the average altitude fault is 1.844 s.

12.3.5.9 Portion of UCAVs Returned on Time

A significant main effect for Number of UCAVs was found $F(2, 28) = 37.66$, ($p \leq 0.0001$). Posthoc Tukey's tests indicated that as the number of UCAVs increased, the portion of UCAVs returned on time significantly decreased. All conditions were significantly different from one another.

12.3.5.10 Simulation Score

The results indicated a significant main effect for Automation $F(1, 14) = 34.74$, ($p \leq 0.0001$). There was a significantly higher UCAV score with HIGH Automation ($X = 335.21$) than with LOW Automation ($X = 104.79$). The results indicate a significant main effect for Number of UCAVs $F(1, 14) = 47.81$, ($p \leq 0.0001$). There was a significantly higher UCAV score for 1 UCAV ($X = 420.00$) than for 2 UCAVs ($X = 272.03$), and controlling 1 UCAV and 2 UCAVs resulted in significantly higher scores than controlling 4 UCAVs ($X = -32.03$).

The main effects are modified by the significant interaction between Automation and Number of UCAVs $F(2, 28) = 10.13$, ($p = 0.0005$) which is illustrated in Fig. 12.7. Posthoc Tukey's tests results reveal that for the LOW Automation condition there is a significant decrease in UCAV score as the number of UCAVs increases from 1 to 4. For the HIGH Automation condition there is a significant decrease in UCAV score when the number of UCAVS increases from 2 to 4. Controlling 4 UCAVs with HIGH Automation significantly improved performance compared to 4 UCAVs with LOW Automation, with an increase in score of 405 points. The score for 4 UCAVs under HIGH Automation was similar to performance with 2 UCAVs under LOW Automation.

12.4 Discussion

The interface and simulation system was designed to allow analysis of uncertainties about UCAV interface design, including: the number of UCAVs a single controller can manage and how proper automation features affect performance in such a complex dynamic mission. The simulation provided an environment in which controller performance could be measured so that the system design and automation issues can be evaluated and analyzed.

The literature indicates that automation can improve as well as degrade human performance in a complex system (Lucas et al. 2001). The ability to control multiple UCAVs will depend on the level of automation and how the operator interacts with the system to complete the mission. Multiple dependent variables related to performance were examined in the simulation and they were useful in providing feedback regarding human performance that can be used for system design input.

12.4.1 Results

For seven of the ten dependent variables there was a significant interaction between the Number of UCAVs and Automation level (first seven dependent variables listed in Table 12.4 and marked with an [a]). Six of those seven dependent variables (dependent variables numbered 1–6 in Table 12.4) showed a pattern of decrease in performance as the number of UCAVs increased under the LOW Automation condition. When conducting the same mission under the HIGH Automation, performance for these six dependent variables, in general, would improve for each Number of UCAVs as compared to LOW Automation. Specifically, performance when controlling 4 UCAVs was not significantly different than when controlling 2 UCAVs for five of the six dependent variables under the HIGH Automation condition. The only dependent variable that showed a difference between 4 and 2 UCAVs was the dependent variable Simulation Score. This dependent variable still indicated significant improvement when using HIGH Automation compared to using LOW Automation. However, this dependent variable is a composite UCAV score that integrates tasks that were both positively and negatively influenced by HIGH Automation. The results of simulation score mirrored the general findings of the individual dependent variables as would be

Table 12.4 Summary of how Automation Levels Affect Performance on the 10 Dependent Variables

HIGH automation improves performance	HIGH automation degrades performance	HIGH automation did not significantly affect performance
1. Correctly identified[a]	7. Portion of tanks destroyed[a]	9. UCAVs returned on time
2. Not identified[a]	8. Incorrectly identified	10. Altitude fault
3. Enemy targets destroyed[a]		
4. Non-radar range faults[a]		
5. Times in range without taking action[a]		
6. simulation score[a]		

[a] significant interactions found

expected, and was a sensitive dependent variable for overall performance. The reader should keep in mind that the subject was trained to understand what the most important tasks were within the mission and the score assigned to each task was designed to match this training.

The dependent variable Portion of Targets Incorrectly Identified and Portion of Enemy Tanks Destroyed were negatively affected under HIGH Automation. Under HIGH Automation the operators may become more complacent in determining that the target is being identified correctly by the system or place too much trust in the automation. This provides a situation of management by consent. When the operator is busy, complacent, or too trusting they may not carefully review the information.

When comparing performance for controlling 4 UCAVs versus 1 UCAV, the results were similar to the comparison between 4 and 2 UCAVs. Under HIGH Automation, four of the dependent variables showed no significant differences between performance for 4 UCAVs versus 1 UCAV. The dependent variables that showed significant differences between these two conditions were Simulation Score and Portion of Targets Not Identified. As already mentioned the dependent variable Simulation Score is composite and shows both positive and negative effects. For Portion of Targets Not Identified, the results were significantly different between 1 and 4 UCAVs because all targets were always identified when subjects only had to control 1 UCAV. With 4 UCAVs to control, the operator had a difficult time identifying all targets. Adding automation significantly improved performance, but perfect identification was not reached.

The increased level of automation under the HIGH Automation condition did have a negative effect on performance for the dependent variable Portion of Targets Incorrectly Identified. A significant main effect indicated that more targets were incorrectly identified under the HIGH Automation condition regardless of the number of UCAVs. A possible explanation for these results is that the operator put too much faith in the ability of the automation to correctly identify targets. After the target was automatically identified the operator may have accepted the identification without checking to determine accuracy because he or she was busy with other tasks.

However, the actual failure rate of automation without any intervention would have been 25%. The average failure rate for HIGH Automation was 6.8%. This means that operators were able to reduce the automation failure rate 18.2% and still free up enough time to improve performance on other dependent variables to optimize his or her overall score. Therefore, the operator may not have been entirely complacent; rather the automation may not have been sufficient to eliminate the failures.

The dependent variable Portion of Enemy Tanks Destroyed was also negatively affected under HIGH Automation when two and four UCAVS were being controlled. This may be due to complacency, level of trust ,or high workload such that operators are not accurately verifying the type of target being destroyed.

12.4.2 Monitoring Task

In this study, one dependent variable was related to monitoring the system status which is the number of times in range doing nothing. There was a significant main effect in controlling 4 UCAVs under the HIGH Automation versus under the LOW Automation condition. The small number of times in range doing nothing under HIGH Automation condition indicates that HIGH Automation condition increased the operator's awareness of the secondary system status. Apparently, enough resources were freed up to allow the operator to attend to these needs. This finding was similar to the research findings conducted by Nathalie et al. (2008); Ruff et al. (2002) and Riley (1996). They suggested that automation reduces workload and frees up resources that the operator can use for monitoring secondary displays and planning future actions.

12.4.3 Control Station Interface Design

Results showed that when subjects controlled only one UCAV their performance was very high for all the dependent variables. For this 1 UCAV condition, performance showed slight improvements under the HIGH Automation condition, but it was never statistically significant. The fact that performance was high when controlling one UCAV indicates that it was not difficult for subjects to carry out tasks using the interface. Changes in the performance were primarily due to an increase in workload as the number of UCAVs being controlled increased, and due to changes in the level of automation. As indicated previously, when controlling four UCAVs under HIGH Automation performance improved significantly reaching levels similar to controlling one UCAV. If the interface was a significant detriment to performance when controlling multiple UCAVs we would not expect to see performance levels similar to levels found when controlling one UCAV. Designing the system to support Mosier's and Skitka's (1996) capabilities of automation aids was successful for providing a system that supported the operator's tasks.

12.5 Conclusions

The command and control of UCAVs requires significant human interaction. There are many research issues that must be investigated to design and develop robust and effective systems. In order to examine human interaction issues in this complex domain it is necessary to develop simulation tools that are flexible and can collect human performance data.

For this project an interface and simulation tool was developed to examine human performance issues when controlling multiple UCAVs. The simulation tool was designed to be adaptable to allow for continued research on a variety of factors related to the control of autonomous vehicles. This study investigated the effects of automation and number of UCAVs being controlled on operator performance during SEAD mission. It was hypothesized that increasing the Number of UCAVs would increase workload and directly affect performance, but differences in performance would be smaller when HIGH Automation was introduced. The results indicate that increasing the number of UCAVs significantly increased workload under LOW Automation. As expected, the increase in workload was reduced when HIGH Automation was introduced. This study illustrates that operators are able to control multiple UCAVs with appropriate automation. The study also lays the foundation and ground rules of the design and development of the UCAV control station interface.

Based on the literature review and research results (Endsley and Kaber 1999; Parasuraman et al. 2000) it is evident that human operators may not perform well with high Automation due to information overload and inappropriate feedback. Therefore, it is important to provide cues in an improved way to keep the humans in the loop. This study presents a systematic structured way of providing feedback based on Mosier and Skitka's (1996) capabilities of automation to actively involve humans in the decision- making process. Subsequent studies repeating the use of these dependent variables would help to determine their reliability and validity.

References

Alberdi E, Strigini L, Povyakalo AA, Ayton P (2009) Why are people's decisions sometimes worse with computer support? Proceedings of the 28th International Conference on Computer safety, reliability and security, Hamburg, Germany, pp18–31

Barnes MJ, Matz MF (1998) Crew simulations for unmanned aerial vehicle (UAV) applications: sustained effects, shift factors, interface issues, and crew size. In: Proceedings of the human factors and ergonomics society 42nd annual meeting, Chicago, IL, October 5–9, pp 143–147

Cummings ML, Bruni S, Mercier S, Mitchell PF (2007) Automation architecture for single operator, multiple UAV command and control. Int CE J 1(2)

Cummings ML, Mitchell PJ (2008) Predicting controller capacity in supervisory control of multiple UAVs. IEEE Trans Syst Man CybernPart A Syst Hum 38:451–460

Dixon SR, Wickens CD (2004) Automation, reliability in unmanned aerial vehicle flight control. In: Vincenzi DA, Mouloua M, Hancock PA (eds) Human performance, situational awareness and automation: current research and trends, pp 205–210

Endsley MR (1987) The application of human factors to the development of expert systems for advanced cockpits. Proc Hum Factors Soc 31st Annu Meet 1387–1392

Endsley MR (1996) Automation and situation awareness. In: Parasuraman R, Mouloua M (eds) Automation and human performance: theory and application, pp 163–181

Endsley MR, Kaber DB (1999) Level of automation effects on performance, situational awareness and workload in a dynamic control task. Ergonomics 42:462–492

Jessie YC, Barnes MJ (2008) Robotics operator performance in a military multi-tasking environment. In: Proceedings of the 3rd ACM/IEEE international conference on human robot interaction, Amsterdam, Netherlands, pp 279–286

Lucas JR, Gallimore, JJ, Prabhala S (2001) Using decision structure to analyze complex semi-autonomous systems. In: Proceedings of the international conference on computer-aided ergonomics and safety, 28 July–2 August 2001, Maui, Hawaii

Meyer J, Feinshreiber L, Parmet Y (2003) Levels of Automation in a Simulated Failure Detection Task. IEEE Int Conf Syst Man Cybern 3:2101–2106

Mooij M, Corker K (2002) Supervisory control paradigm: limitations in applicability to advanced air traffic management systems. In: Proceedings of IEEE digital avionics system conference, vol 1, pp IC3.1–IC3.8

Mosier KL, Skitka LJ (1996) Human decision makers and automated decision aids: made for each other? In: Mouloua M and Parasuraman R (eds) Human performance in automated systems: current research and trends, pp 201–220

Nathalie P, Xavier N, David H, Eric S (2008) Physiological investigation of vigilance decrement: Boredom or cognitive fatigue? Physiol Behav 93(1–2):369–378

arasuraman P R, Sheridan TB, Wickens CD (2000) A model for type and level of human interaction with automation. IEEE Trans Syst Man Cybern 30(3):286–297

Parasuraman R, Hancock PA (2008) Mitigating the adverse effects of workload, stress, and fatigue with adaptive automation. In: Hancock P, Szalma JL (eds) Performance under Stress, pp 45–58

Riley V 1996 Operator reliance on automation: theory and data. In: Parasuraman R, Mouloua M (eds) Automation and human performance: theory and applications, pp 19–35

Ruff HA, Narayanan S, Draper MH (2002) Human interaction with levels of automation and decision-aid fidelity in the supervisory control of multiple unmanned air vehicles. Presence 11(4):335–351

Sarter NB, Woods DD (1995) How in the world did we get into that mode? mode error and awareness in supervisory control. Hum Factors 37(1):5–19

Sheridan TB (1980) Computer control and human alienation. Technol Rev, pp 60–73

Taylor RM (2002) Capability, cognition and autonomy. In: Proceedings of RTO Human Factors and Medicine Panel (HFM) Symposium, Warsaw, Poland, October 7–9

Warm JS, Matthews G, Finomore JR VS (2008) Vigilance, workload, and stress. In: Hancock P, Szalma J.L (eds) Performance under stress, pp 115–142

Wickens CD (1999) Automation in air traffic control: the human performance Issues. In: Scerbo M, Mouloua M (eds) Automation technology and human performance: current research and trends, pp 2–10

Index

L. Rothrock and S. Narayanan (eds.), *Human-in-the-Loop Simulations*,
DOI: 10.1007/978-0-85729-883-6, © Springer-Verlag London Limited 2011